

УДК 004.052.32+681.518.5

## ВЗВЕШЕННЫЕ КОДЫ БОУЗА–ЛИНА В СХЕМАХ ВСТРОЕННОГО КОНТРОЛЯ НА ОСНОВЕ ЛОГИЧЕСКОЙ КОРРЕКЦИИ СИГНАЛОВ

© 2025 г. Е. И. Елина<sup>а, \*</sup>, Д. В. Ефанов<sup>а, б, \*\*</sup>

<sup>а</sup>Санкт-Петербургский политехнический ун-т Петра Великого, Санкт-Петербург, Россия

<sup>б</sup>Российский университет транспорта (МИИТ), Москва, Россия

\*e-mail: eseniya-elina@mail.ru

\*\*e-mail: TrES-4b@yandex.ru

Поступила в редакцию 22.03.2024 г.

После доработки 27.09.2024 г.

Принята к публикации 13.01.2025 г.

Исследуются возможности применения взвешенных кодов Боуза–Лина при синтезе схем встроенного контроля на основе логической коррекции сигналов. Установлены зависимости числа вариантов взвешивания при построении рассматриваемого класса кодов от числа  $m$  информационных символов. С увеличением значения числа  $m$  количество способов взвешивания существенно возрастает, что позволяет строить большое количество кодов и, как следствие, иметь большую вариативность при построении самопроверяемого устройства. Установлены закономерности, проявляющиеся при изменении числа информационных символов, которые связывают число вариантов взвешивания при построении кода. Число способов синтеза схем встроенного контроля на базе логической коррекции сигналов с применением взвешенных кодов Боуза–Лина факториально зависит от числа выделяемых информационных символов среди рабочих сигналов объекта диагностирования. Приведенные примеры демонстрируют эффективность использования взвешенных кодов Боуза–Лина при синтезе схем встроенного контроля по сравнению с применением традиционного метода, основанного на дублировании. Результаты исследования целесообразно учитывать при разработке самопроверяемых цифровых устройств, а также при развитии систем их автоматизированного проектирования.

**Ключевые слова:** самопроверяемое дискретное устройство, схема встроенного контроля, контроль вычислений, взвешенный код с суммированием, логическая коррекция сигналов, обеспечение самопроверяемости

DOI: 10.31857/S0002338825010047 EDN: AGWTRE

## WEIGHT-BASED BOSE–LIN CODES IN CONCURRENT ERROR-DETECTION CIRCUIT BASED ON BOOLEAN SIGNALS CORRECTION

Y. I. Yelina<sup>а, \*</sup>, D. V. Efanov<sup>а, б, \*\*</sup>

<sup>а</sup>Peter the Great Saint Petersburg Polytechnic University, St. Petersburg, Russia

<sup>б</sup>Russian University of Transport, Moscow, Russia

\*e-mail: eseniya-elina@mail.ru

\*\*e-mail: TrES-4b@yandex.ru

The paper explores the possibilities of using weight-based Bose–Lin codes in the synthesis of concurrent error-detection circuit based on Boolean signals correction. The dependences of the number of weighing options in the construction of the considered class of codes on the number  $m$  of data symbols are established. It is shown that with an increase in the value of the number  $m$ , the number of weighing methods increases significantly, which makes it possible to build a large number of codes and, as a result, have greater variability in the construction of a self-checking device. The patterns that appear when the number of data symbols changes are established, linking the number of weighing options when building the code. It is shown that the number of ways to synthesize concurrent error-detection circuits based on Boolean signals correction using weight-based Bose–Lin codes factorially depends on the number of allocated data symbols among the data signals of the diagnostic object. The examples given in the paper demonstrate the effectiveness of using weight-based Bose–Lin codes in the synthesis of concurrent error-detection circuit compared to using the traditional method based on duplication. It is advisable to take the results of the study into account when developing self-checking digital devices, as well as when developing their computer-aided design systems.

**Keywords:** self-checking discrete device; concurrent error-detection circuit; calculation control; weight-based sum code; Boolean signals correction; self-checking provision

**Введение.** Устройства и современные системы критического применения в промышленности и на транспорте должны быть высоконадежными и безопасными. В этой связи первостепенной оказывается задача своевременного обнаружения неисправностей в их структурах и вызываемых тем самым ошибок в вычислениях [1–4]. Решение данной задачи связано с наделением структур составляющих, блоков и отдельных подсистем свойствами контролепригодности, самопроверяемости, отказоустойчивости и безопасного поведения при отказах [5].

Обнаружение неисправностей и их проявлений осуществляется в том числе с помощью самопроверяемых схем встроенного контроля (СВК) [6, 7]. Они могут быть синтезированы по различным принципам большим количеством способов. Широко зарекомендовали себя две типовые структуры организации СВК. Первая из них основана на использовании свойств кодов с повторением и предполагает дублирование исходного устройства и последующее сравнение сигналов на одноименных выходах с помощью самопроверяемого компаратора [8]. Вторая структура организации СВК подразумевает контроль вычислений на выходах исходного устройства по коду паритета: рассчитывается значение свертки по модулю  $M = 2$  для сигналов с выходов контролируемого объекта и сравнивается в СВК со значением функции паритета, вычисленной по значениям входных воздействий на контролируемый объект [9]. У обеих структур есть достоинства и недостатки. Например, в структуре на основе дублирования обнаруживаются любые сочетания искажений на выходах исходного устройства, однако для ряда устройств сложно и даже невозможно обеспечить полную самопроверяемость СВК за счет особенностей структуры компаратора [10]. При этом показатели структурной избыточности для устройства с СВК оказываются существенными, что на практике может приводить к построению устройств, показатели сложности реализации которых в 3–4 раза выше, чем исходного объекта. Кроме того, при дублировании нельзя эффективно учитывать особенности моделей неисправностей, относительно которых и строятся устройства со свойствами самопроверяемости (например, учитывать покрытие любых возникающих неисправностей или наиболее вероятных), что в конечном итоге приводит к использованию излишней избыточности. У структуры, основанной на применении кода паритета, недостаток связан с тем, что любая ошибка с четной кратностью на выходах исходного устройства обнаружена не будет, что потребует специальных схемотехнических методов для построения полностью самопроверяемых структур [6, 9, 11, 12].

По обеим типовым структурам, базирующимся на кодах с повторением и паритета, можно синтезировать одну СВК. Это не позволяет влиять на показатели эффективности реализации самопроверяемых устройств, например структурной избыточности, тестопригодности и пр.

Отмеченный выше недостаток нивелируется использованием структуры с логической коррекцией сигналов (ЛКС) в СВК, предложенной, по всей видимости, профессорами Сапожниковыми в работе [13] для контроля вычислений на основе свертки по модулю  $M = 2$  и последующего преобразования единственной контрольной функции в самодвойственную. Применение ЛКС дает возможность построения большого числа СВК с контролем вычислений по различным диагностическим признакам [14, 15].

В научной литературе уделено большое внимание синтезу СВК на базе ЛКС с применением для контроля двоичных неразделимых кодов и принадлежности формируемых функций классу самодвойственных [16–20]. Однако организации СВК на основе ЛКС с применением двоичных делимых кодов посвящено не так много работ. Среди них отметим статью [21], в которой предложено использовать коды Бергера [22] с числом информационных символов  $m = 3$  для контроля вычислений по группам выходов контролируемых устройств, а также работу [23], где этот способ обобщен для применения произвольных кодов Бергера и их известной модификации — кода с суммированием в кольце вычетов по заданному модулю  $M$  (остаточных кодов [24, 25]). Предложено рассматривать в качестве базовых кодов для построения СВК на основе ЛКС коды с суммированием в кольце вычетов по модулям  $M = 4$  и 8 (они известны в мире как коды Боуза–Лина [26]).

Настоящая статья посвящена развитию теории синтеза СВК на основе ЛКС с применением свойств модифицированных кодов с суммированием — *взвешенных кодов Боуза–Лина*. Иссле-

дования показывают, что их использование позволяет синтезировать для исходных устройств большое количество СВК с разнообразными показателями эффективности.

**1. Взвешенные коды Боуза–Лина.** Коды Боуза–Лина, как и другие модульные коды, за счет своих особенностей применяются при синтезе устройств и систем, где требуется обнаружение неисправностей и ошибок в вычислениях [27]. К кодам Боуза–Лина относят делимые коды, для которых в контрольных векторах записывается значение наименьшего неотрицательного вычета по модулю  $M \in \{4, 8, 12\}$  суммы значащих разрядов в информационных векторах (веса информационного вектора) [26]. Характеристики обнаружения ошибок кодами Боуза–Лина на случай их возникновения исключительно в информационных векторах<sup>1</sup> изучены в работах [28–30], во всех разрядах кодовых слов<sup>2</sup> – в работах [31, 32]. Интересным свойством кодов Боуза–Лина является то, что доля необнаруживаемых ими ошибок в информационных векторах от общего числа ошибок, возникающих в информационных векторах, не зависит от числа  $m$  и является постоянной величиной [29]. Кроме того, кодами Боуза–Лина обнаруживаются любые однонаправленные (монотонные) ошибки в информационных векторах, за исключением 100% ошибок кратностью  $d = jM$ ,  $j = 1, 2, \dots, \lfloor m/M \rfloor$ , где  $m$  – число информационных символов. Эти особенности определяют широкие возможности применения данного класса кодов с суммированием при синтезе устройств и систем с обнаружением неисправностей.

Среди кодов Боуза–Лина в задачах синтеза самопроверяемых цифровых устройств наибольшее распространение получили коды, для которых при построении зафиксирован модуль  $M = 4$  или  $M = 8$ . Такие коды имеют соответственно  $k = 2$  и  $3$  проверочных символа, а при рассмотрении полного множества кодовых слов наблюдается формирование и полного множества контрольных векторов для двух и трех разрядов, что позволяет легко обеспечивать самопроверяемость тестера данного кода [33].

Исследования [34] показывают, что классические коды Боуза–Лина – это только подмножество кодов из множества так называемых взвешенных кодов с суммированием в кольце вычетов по модулю  $M$ . При построении таких кодов разрядам  $f_i$  информационных векторов предварительно приписывают некоторые весовые коэффициенты  $w_i$ ,  $i = \overline{1, M-1}$ , а затем получают суммарное значение весовых коэффициентов значащих разрядов информационного вектора в кольце вычетов по выбранному модулю – число

$$W_M = \left( \sum_{i=1}^m w_i f_i \right) \pmod{M},$$

и записывают его в контрольный вектор. Введем для взвешенных кодов Боуза–Лина следующее обозначение:  $WS(m, k, M)$  коды<sup>3</sup>. Число проверочных символов определяется как  $k = \lceil \log_2 M \rceil$ , число  $WS(m, k, M)$  кодов –

$$N_{WS(m, k, M)} = (M - 1)^m. \quad (1.1)$$

К примеру, при  $M = 4$  и  $m = 4$  формула (1.1) дает:

$$N_{WS(4, 2, 4)} = (4 - 1)^4 = 81 \text{ код.}$$

Если рассматриваются характеристики обнаружения ошибок кодами на полном множестве их кодовых слов, то оказывается несущественным, какой из разрядов информационного вектора взвешен каким числом, важными являются только значения самих весовых коэффициентов. Поэтому  $WS(m, k, M)$  кодов с различными характеристиками обнаружения ошибок гораздо меньше, чем общее их количество, получаемое по формуле (1.1).

<sup>1</sup> В такой постановке задача актуальна в тех приложениях кодов Боуза–Лина, в которых устройства вычисления информационных и проверочных символов физически разделены.

<sup>2</sup> В данной постановке задача актуальна в тех приложениях кодов Боуза–Лина, в которых одно из вычислительных устройств влияет на значения информационных и проверочных символов.

<sup>3</sup>  $WS$  – от англ. weight-based sum code.



Таблица 2. Число  $WS(m, 3, 8)$  кодов

$m$	Число $WS(m, 3, 8)$ кодов
1	$N_1 = M - 1 = 8 - 1$
2	$N_2 = N_1 + 6 + 5 + 4 + 3 + 2 + 1 = 28$
3	$N_3 = N_2 + (28 - 7) + ((28 - 7) - 6) + (((28 - 7) - 6) - 5) + (((((28 - 7) - 6) - 5) - 4) +$ $+ ((((((28 - 7) - 6) - 5) - 4) - 3) + (((((((28 - 7) - 6) - 5) - 4) - 3) - 2) = 28 + 21 + 15 +$ $+ 10 + 6 + 3 + 1 = 84$
4	$N_4 = N_3 + (84 - 28) + ((84 - 28) - 21) + (((84 - 28) - 21) - 15) + ((((((84 - 28) - 21) -$ $- 15) - 10) + (((((((84 - 28) - 21) - 15) - 10) - 6) + ((((((((((84 - 28) - 21) - 15) - 10) -$ $- 6) - 3) = 84 + 56 + 35 + 20 + 10 + 4 + 1 = 210$
5	$N_5 = 210 + 126 + 70 + 35 + 15 + 5 + 1 = 462$
...	...
10	$N_{10} = 5005 + 2002 + 715 + 220 + 55 + 10 + 1 = 8008$

Выражение (1.3) проиллюстрируем на фрагменте треугольника Паскаля (рис. 2). В нем рассмотрение случая  $m = 1$  начинается со строки с номером  $M - 1 = 4 - 1 = 3$  (нумерация строк – сверху вниз, начиная с 0), далее – с увеличением номера строки – по диагонали. Для  $m = 2$  – строка № 4,  $m = 3$  – строка № 5 и т.д. Можно заметить, что последовательность чисел формирует так называемые треугольные числа (последовательность A000217 из работы [35]).

Аналогично, но несколько сложнее рассматривается и число способов построения  $WS(m, 3, 8)$  кодов. Если  $m = 1$ , то при подсчете наименьшего неотрицательного вычета весовой коэффициент  $w_1$  может быть выбран из множества  $\{1, 2, \dots, 7\}$ . При  $m = 2$  появляется второй разряд в информационном векторе, который также может быть взвешен числами из множества  $\{1, 2, \dots, 7\}$ . На рис. 3 иллюстрируется процесс увеличения числа вариантов взвешивания при увеличении числа разрядов в информационном векторе.

Анализ числа вариантов взвешивания при построении  $WS(m, 3, 8)$  кодов приводит к табл. 2, по которой также можно установить закономерность. Читатель вправе обратить внимание на треугольник Паскаля, рассмотренный на рис. 2, и заметить, что число способов взвешивания при  $m = 1$  соответствует второму члену в строке с номером  $M - 1 = 8 - 1 = 7$ , при  $m = 2$  – третьему члену в строке с номером 8 и т.д. Суммируя числа по диагонали над числом способов взвешивания при  $m - 1$  с данным числом, получаем общее число вариантов взвешивания для  $m$  и т. д. Это последовательность A000579 из [35].

Из анализа числа способов взвешивания при построении  $WS(m, 3, 8)$  кодов следует, что

$$N_{WS(m, 3, 8)} = C_{m+6}^m = \frac{(m+6)!}{m!6!} = \frac{(m+1)(m+2) \dots (m+6)}{720}. \quad (1.4)$$

Свойства  $WS(m, k, M)$  кодов будут определяться конкретным сочетанием весовых коэффициентов. Здесь подробно рассматривать их не будем. Некоторые примеры можно найти в работе [34].

Обратимся к особенностям построения схемы встроенного контроля на основе взвешенных кодов Боуза–Лина.

**2. Структура организации схем встроенного контроля на базе логической коррекции сигналов с применением взвешенных кодов Боуза–Лина.** На рис. 4

изображена структура организации СВК на основе ЛКС [7, 10, 16]. В ней исходным устройством, или объектом диагностирования, является блок  $F(x)$ , снабженный  $t$  входами и  $n$  выходами, на которых реализуются булевы функции  $f_1, f_2, \dots, f_n$ . Для организации контроля вычислений организуется СВК. В ней использован принцип ЛКС, подразумевающий возможность преобразования функций  $f_1, f_2, \dots, f_n$  в функции  $h_1, h_2, \dots, h_n$ , которые об-

$w_2$	$w_1$
1	1
1	2
1	3
1	4
1	5
1	6
1	7
2	2
2	3
2	4
2	5
2	6
2	7
3	3
3	4
3	5
3	6
3	7
4	4
4	5
4	6
4	7
5	5
5	6
5	7
6	6
6	7
7	7

Рис. 3. Варианты взвешивания при построении  $WS(m, 3, 8)$  кодов

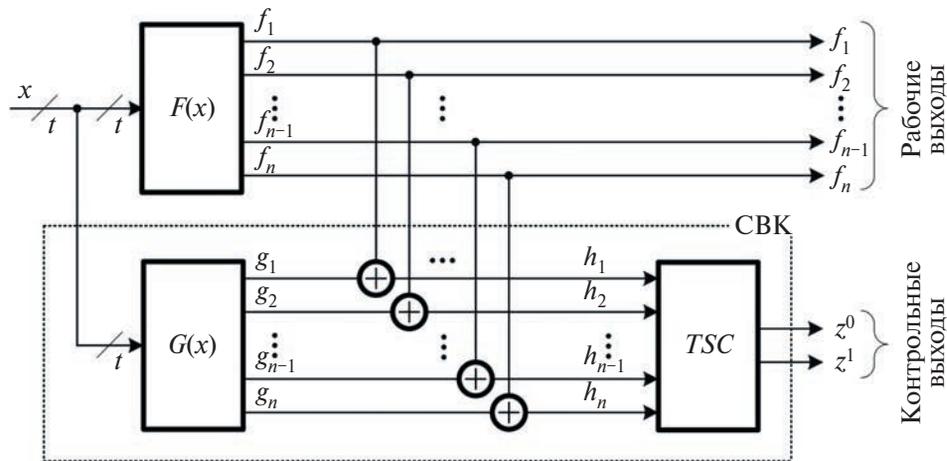


Рис. 4. Структура организации СВК на основе ЛКС

ладают определенными диагностическими свойствами. Преобразования осуществляются по формуле  $h_i = f_i \oplus g_i, i = 1, n$ , где  $g_i$  –  $i$ -я функция логической коррекции, вычисляемая блоком контрольной логики  $G(x)$ . Для контроля вычислений по выбранному диагностическому признаку устанавливается тестер  $TSC$  (totally self-checking checker) [6, 7].

При организации СВК на основе ЛКС с контролем вычислений по  $WS(m, k, M)$  кодам функции  $h_1, h_2, \dots, h_n$  формируются таким образом, чтобы вектор их значений на каждой входной комбинации  $\langle h_n h_{n-1} \dots h_2 h_1 \rangle$  принадлежал выбранному коду. Контроль соответствия информационных и проверочных символов друг другу осуществляют с помощью самопроверяемого тестера  $WS(m, k, M)$  кода.

Достоинством ЛКС является то, что для одного объекта диагностирования при выбранном способе организации СВК и установленном  $WS(m, k, M)$  коде существует большое количество вариантов формирования функций  $h_1, h_2, \dots, h_n$ . Это дает возможность синтеза большого количества СВК и, как следствие, оптимизации показателей их эффективности. Определим, какое количество вариантов построения СВК существует при использовании рассматриваемых  $WS(m, 2, 4)$  и  $WS(m, 3, 8)$  кодов. Для этого установим один из вариантов синтеза СВК на основе ЛКС с применением разделимых кодов, описанный в работе [23]. Подразумевается коррекция только тех рабочих функций объекта диагностирования, которые участвуют в формировании проверочных символов кода Боуза–Лина. В этом случае доопределение функций логической коррекции сигналов, формирующих проверочные символы, является однозначным.

Число способов выбора  $k$  преобразуемых функций из  $n = m + k$  определяется величиной  $C_{m+k}^k$ . Число вариантов размещения  $k$  проверочных символов в контрольном векторе равно  $P_k$ . Так как каждому не преобразуемому символу будет приписан какой-либо весовой коэффициент, то имеет значение последовательность расположения информационных символов в информационном векторе, определяемая числом перестановок из  $m$  элементов – числом  $P_m$ . Отсюда получаем общее количество вариантов синтеза СВК:

$$N_{WS(m, k, M)}^{CED} = C_{m+k}^k P_m P_k = \frac{(m+k)!}{k!m!} m!k! = (m+k)! \quad (2.1)$$

Поскольку  $k = \lceil \log_2 M \rceil$  в конечном выражении (2.1) значение  $M$  не присутствует. Для рассматриваемых  $WS(m, 2, 4)$  и  $WS(m, 3, 8)$  кодов формула (2.1) дает  $N_{WS(m, 2, 4)}^{CED} = (m+2)!$  и  $N_{WS(m, 3, 8)}^{CED} = (m+3)!$  число способов синтеза СВК. С учетом того, что для каждого значения  $m$  возможно различное количество вариантов взвешивания при построении кода, получаем следующее количество способов синтеза СВК на основе  $WS(m, 2, 4)$  и  $WS(m, 3, 8)$  кодов:

$$N_{WS(m, 2, 4)}^{CED*} = N_{WS(m, 2, 4)} N_{WS(m, 2, 4)}^{CED} = C_{m+2}^m (m+2)! \quad (2.2)$$

$$N_{WS(m, 3, 8)}^{CED*} = N_{WS(m, 3, 8)} N_{WS(m, 3, 8)}^{CED} = C_{m+6}^m (m+6)! \quad (2.3)$$

**Таблица 3.** Число вариантов синтеза СВК на основе ЛКС с применением взвешенных кодов Боуза–Лина

$m$	$N_{WS(m, 2, 4)}^{CED*}$	$N_{WS(m, 3, 8)}^{CED*}$	$\zeta$
3	1 200	30 481 920	25 401.6
4	10 800	762 048 000	70 560
5	105 840	18 441 561 600	174 240
6	1 128 960	442 597 478 400	392 040
7	13 063 680	10 685 567 692 800	817 960
8	163 296 000	261 796 408 473 600	1 603 201.6
9	2 195 424 000	6 544 910 211 840 000	2 981 160
10	31 614 105 600	167 549 701 423 104 000	5 299 840
11	485 707 622 400	4 401 987 610 116 100 000	9 063 040
12	7 933 224 499 200	118 853 665 473 135 000 000	14 981 760
13	137 305 808 640 000	3 300 474 864 292 430 000 000	24 037 401.6
14	2 510 734 786 560 000	94 299 281 836 926 600 000 000	37 558 440
15	48 373 490 221 056 000	2 772 398 886 005 640 000 000 000	57 312 360
16	979 563 176 976 384 000	83 865 066 301 670 600 000 000 000	85 614 760
17	20 801 312 169 910 300 000	2 609 683 533 740 220 000 000 000 000	125 457 640
18	462 251 381 553 562 000 000	83 509 873 079 687 100 000 000 000 000	180 659 001.6
19	10 729 097 856 059 000 000 000	2 747 035 298 673 920 000 000 000 000 000	256 036 000
20	259 644 168 116 627 000 000 000	92 849 793 095 178 400 000 000 000 000 000	357 604 000

В табл. 3 приведены результаты расчетов по формулам (2.2) и (2.3) для малых значений  $m$ . В последней графе вычислен коэффициент  $\zeta$ , показывающий, во сколько раз число вариантов построения СВК с применением  $WS(m, 3, 8)$  кода больше числа вариантов построения СВК с применением  $WS(m, 2, 4)$  кода при одинаковых значениях  $m$ .

Отметим, что число вариантов построения СВК на основе ЛКС по взвешенному коду Боуза–Лина существенно увеличивается при увеличении  $m$  даже на 1:

$$\begin{aligned} \delta_{m+1, m}^{M=4} &= \frac{N_{WS(m+1, 2, 4)}^{CED*}}{N_{WS(m, 2, 4)}^{CED*}} = \frac{C_{(m+1)+2}^{(m+1)} ((m+1)+2)!}{C_{m+2}^m (m+2)!} = \frac{\frac{(m+3)!}{(m+1)!2!} (m+3)!}{\frac{(m+2)!}{m!2!} (m+2)!} = \\ &= \frac{(m+2)!(m+3)}{m!(m+1)} \frac{(m+2)!(m+3)}{(m+2)!(m+2)!} = \frac{(m+3)^2}{m+1} = m+5 + \frac{4}{m+1}; \end{aligned} \tag{2.4}$$

$$\begin{aligned} \delta_{m+1, m}^{M=8} &= \frac{N_{WS(m+1, 3, 8)}^{CED*}}{N_{WS(m, 3, 8)}^{CED*}} = \frac{C_{(m+1)+6}^{(m+1)} ((m+1)+6)!}{C_{m+6}^m (m+6)!} = \frac{\frac{(m+7)!}{(m+1)!6!} (m+7)!}{\frac{(m+6)!}{m!6!} (m+6)!} = \\ &= \frac{(m+6)!(m+7)}{m!(m+1)6!} \frac{(m+6)!(m+7)}{(m+6)!(m+6)!} = \frac{(m+7)^2}{m+1} = m+13 + \frac{36}{m+1}. \end{aligned} \tag{2.5}$$

В табл. 4 представлена зависимость показателей  $\delta_{m+1, m}^{M=4}$  и  $\delta_{m+1, m}^{M=8}$  от числа  $m$ . С ростом  $m$  значения данных показателей также возрастают. Как следует из формул (2.4) и (2.5), рост является линейным.

Таблица 4. Зависимость показателей  $\delta_{m+1,m}^{M=4}$  и  $\delta_{m+1,m}^{M=8}$  от числа  $m$ 

$m$	$\delta_{m+1,m}^{M=4}$	$\delta_{m+1,m}^{M=8}$
3	9	25
4	9.8	24.2
5	10.667	24
6	11.571	24.143
7	12.5	24.5
8	13.444	25
9	14.4	25.6
10	15.364	26.273
11	16.333	27
12	17.308	27.769
13	18.286	28.571
14	19.267	29.4
15	20.25	30.25
16	21.235	31.118
17	22.222	32
18	23.211	32.895
19	24.2	33.8
20	25.190	34.714

Формулы (2.4) и (2.5) могут быть обобщены на случай рассмотрения кодов с  $m$  и  $m + p$  информационными символами:

$$\begin{aligned}
\delta_{m+p,m}^{M=4} &= \frac{N_{WS(m+p,2,4)}^{CED*}}{N_{WS(m,2,4)}^{CED*}} = \frac{C_{(m+p)+2}^{(m+p)} ((m+p)+2)!}{C_{m+2}^m (m+2)!} = \frac{\frac{((m+p)+2)!}{(m+p)!2!} ((m+p)+2)!}{\frac{(m+2)!}{m!2!} (m+2)!} = \\
&= \frac{\frac{(m+2)!((m+2)+1)((m+2)+1) \cdot \dots \cdot ((m+2)+p)}{m!(m+1)(m+2) \cdot \dots \cdot (m+p)2!}}{\frac{(m+2)!}{m!2!} (m+2)!} \times \\
&\times \frac{(m+2)!((m+2)+1)((m+2)+1) \cdot \dots \cdot ((m+2)+p)}{\frac{(m+2)!}{m!2!} (m+2)!} = \\
&= \frac{\left( ((m+2)+1)((m+2)+1) \cdot \dots \cdot ((m+2)+p) \right)^2}{(m+1)(m+2) \cdot \dots \cdot (m+p)} = \frac{\left( ((m+2)+p)^p \right)^2}{(m+p)^p}; \\
\delta_{m+p,m}^{M=8} &= \frac{N_{WS(m+p,3,8)}^{CED*}}{N_{WS(m,3,8)}^{CED*}} = \frac{C_{(m+p)+6}^{(m+p)} ((m+p)+6)!}{C_{m+6}^m (m+6)!} = \frac{\frac{((m+p)+6)!}{(m+p)!6!} ((m+p)+6)!}{\frac{(m+6)!}{m!6!} (m+6)!} = \\
&= \frac{\frac{(m+6)!((m+6)+1)((m+6)+1) \cdot \dots \cdot ((m+6)+p)}{m!(m+1)(m+2) \cdot \dots \cdot (m+p)6!}}{\frac{(m+6)!}{m!6!} (m+6)!} \times
\end{aligned} \tag{2.6}$$

$$\begin{aligned}
 & \times \frac{(m+6)!((m+6)+1)((m+6)+1) \cdot \dots \cdot ((m+6)+p)}{\frac{(m+6)!}{m!6!}(m+6)!} = \\
 & = \frac{(((m+6)+1)((m+6)+1) \cdot \dots \cdot ((m+6)+p))^2}{(m+1)(m+2) \cdot \dots \cdot (m+p)} = \frac{(((m+6)+p)^p)^2}{(m+p)^p}.
 \end{aligned} \tag{2.7}$$

В формулах (2.6) и (2.7) символом  $x^p$  обозначена убывающая факториальная степень  $x^p = x! / (x-p)!$  [36].

С практической точки зрения интерес также представляет рассмотрение случая сравнения числа вариантов построения СВК на основе ЛКС с применением  $WS(m+1, 2, 4)$  и  $WS(m, 3, 8)$  кодов. Для таких кодов общее число разрядов в кодовых словах одинаково, и их можно применить при синтезе СВК для объектов диагностирования с одинаковым числом выходов. Пользуясь (2.2) и (2.3), запишем отношение числа вариантов построения СВК на основе ЛКС и  $WS(m, 3, 8)$  и  $WS(m+1, 2, 4)$  кодов:

$$\begin{aligned}
 \varepsilon &= \frac{N_{WS(m, 3, 8)}^{CED*}}{N_{WS(m, 2, 4)}^{CED*}} = \frac{C_{m+6}^m (m+6)!}{C_{m+2}^m (m+2)!} = \frac{\frac{(m+6)!}{m!6!} (m+6)!}{\frac{(m+2)!}{m!2!} (m+2)!} = \\
 &= \frac{\frac{((m+2)!)^2 ((m+3)(m+4)(m+5)(m+6))^2}{m!6!}}{\frac{((m+2)!)^2}{m!2!}} = \frac{((m+3)(m+4)(m+5)(m+6))^2}{360}.
 \end{aligned} \tag{2.8}$$

Из формулы (2.8) становится ясно, что число вариантов построения СВК на базе ЛКС при применении  $WS(m, 3, 8)$  кодов существенно выше, чем аналогичный показатель при использовании  $WS(m+1, 2, 4)$ . Обратите внимание на то, как стремительно растет значение показателя  $\varepsilon$  с ростом значения  $m$  (см. табл. 5).

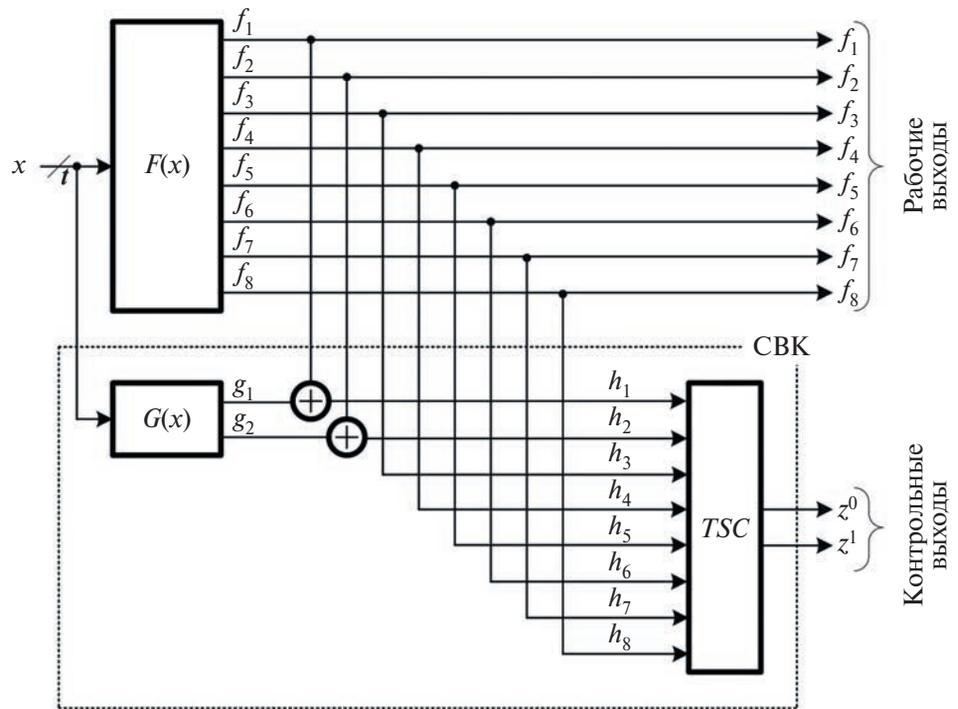
Приведенные здесь результаты исследований говорят о том, что даже при малых значениях  $m$  можно построить внушительное количество СВК на основе ЛКС с помощью взвешенных кодов Боуза–Лина, что определяет и возможности синтеза полностью самопроверяемых структур с различными показателями эффективности. Следует также еще раз подчеркнуть, что нами рассмотрен способ организации СВК, при котором доопределение функций логической коррекции сигналов однозначно. Существуют и иные способы синтеза СВК на основе ЛКС, например при преобразовании всех рабочих функций объекта диагностирования [37]. Число способов синтеза СВК на базе взвешенных кодов Боуза–Лина при неоднозначном доопределении функций логической коррекции сигналов значительно выше, чем при однозначном доопределении.

**3. Синтез схемы встроенного контроля на основе логической коррекции сигналов с применением взвешенного кода Боуза–Лина.** 3.1 Синтез схем встроенного контроля на базе кода с модулем  $M = 4$ . Приведем пример синтеза СВК для устройства  $F(x)$ , снабженного четырьмя входами и восемью выходами, на основе ЛКС с применением  $WS(6, 2, 4)$  кода с массивом весовых коэффициентов  $[w_8, w_7, w_6, w_5, w_4, w_3] = [1, 1, 1, 2, 2, 2]$ . Для этого используем методику, описанную в работе [23], для преобразования в СВК значений двух рабочих функций.

Устройство с СВК представлено на рис. 5. Основной задачей синтеза является получение описания блока контрольной логики  $G(x)$ , поскольку элементы преобразования и тестер  $WS(6, 2, 4)$  кода (тестер включает в себя кодер и компаратор, образованный одним модулем сжатия парафазных сигналов  $TRC$  (two-rail checker) [38]) имеют типовые структуры. Кодеры кодов с суммированием удобно строить на элементной базе, включающей в себя полные

**Таблица 5.** Зависимость показателя  $\epsilon$  от числа  $m$

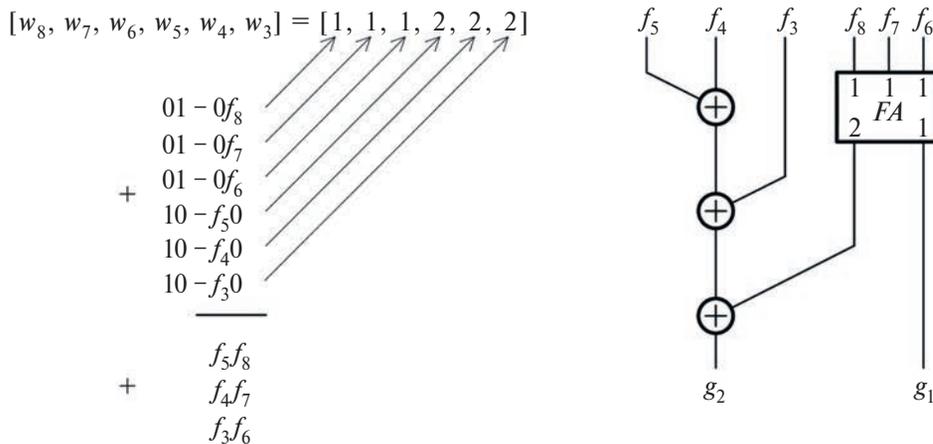
$m$	$\epsilon$
3	25 401.6
4	70 560
5	174 240
6	392 040
7	817 960
8	1 603 201.6
9	2 981 160
10	5 299 840
11	9 063 040
12	14 981 760
13	24 037 401.6
14	37 558 440
15	57 312 360
16	85 614 760
17	125 457 640
18	180 659 001.6
19	256 036 000
20	357 604 000



**Рис. 5.** Структура организации СВК на основе ЛКС для 8-выходного устройства с применением  $WS(6, 2, 4)$  кода

(a)

(б)



**Рис. 6.** Кодер  $WS(6, 2, 4)$  кода: *a* – схема построения, *б* – структура

сумматоры, полусумматоры и четвертьсумматоры (или сумматоры по модулю  $M = 2$ )<sup>4</sup> [39]. На рис. 6 приведена структура кодера  $WS(6, 2, 4)$  кода, синтезированная с помощью данной элементной базы<sup>5</sup>.

Изначально сигналы на выходах блоков СВК не определены. Требуется получить описание работы СВК и значения функций  $h_1, h_2, \dots, h_8$  на каждой входной комбинации. Значения стар-

<sup>4</sup> Полные сумматоры обозначены на рисунках как *FA* (от англ. full-adder), полусумматоры – как *HA* (от англ. half-adder). У входов и выходов сумматоров и полусумматоров указаны значения весов разрядов. Такие обозначения используются в ряде изданий, например в работе [26].

<sup>5</sup> Здесь и далее кодеры синтезированы по методике, подразумевающей уменьшение числа первичных складываемых двоичных чисел, соответствующих весовым коэффициентам, за счет группировки чисел, для сложения которых не требуется операция переноса (у таких чисел в разрядах складываются 0 и значение конкретного информационного символа). Далее кодер синтезирован в виде устройства сложения нескольких двоичных чисел.

**Таблица 6.** Сигналы на линиях СВК на основе ЛКС с применением  $WS(6, 2, 4)$  кода с массивом весовых коэффициентов [1, 1, 1, 2, 2, 2]

$x_4$	$x_3$	$x_2$	$x_1$	$f_8$	$f_7$	$f_6$	$f_5$	$f_4$	$f_3$	$f_2$	$f_1$	$h_8$	$h_7$	$h_6$	$h_5$	$h_4$	$h_3$	$h_2$	$h_1$	$g_2$	$g_1$
0	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0	0
0	0	0	1	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	1	0	1
0	0	1	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0
0	0	1	1	1	1	0	1	0	0	0	1	1	1	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	1	1	0	0	1	0	0	1	1	1	0	0	1	0	0
0	1	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	1	1
0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0	0
0	1	1	1	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	1	0	1
1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0	1	1	0	1
1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	1	1	0
1	0	1	0	1	1	0	1	0	1	0	1	1	1	0	1	0	1	1	0	1	1
1	0	1	1	0	1	0	1	1	0	1	0	0	1	0	1	1	0	0	1	1	1
1	1	0	0	0	1	0	1	0	0	1	1	0	1	0	1	0	0	1	1	0	0
1	1	0	1	1	1	1	0	0	1	1	0	1	1	1	0	0	1	0	1	1	1
1	1	1	0	1	1	0	0	1	0	0	0	1	1	0	0	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1

ших разрядов кодового слова  $\langle h_8 h_7 \dots h_2 h_1 \rangle$  получаются однозначно:  $h_3 = f_3$ ,  $h_4 = f_4$ , ...,  $h_8 = f_8$ . Далее по правилам построения  $WS(6, 2, 4)$  кода получают значения чисел  $W_M$  для каждого информационного вектора и записывают в вектор  $\langle h_2 h_1 \rangle$  соответствующее двоичное число. Значения функций коррекции получают по формулам  $g_2 = h_2 \oplus f_2$ ,  $g_1 = h_1 \oplus f_1$ . Описание работы СВК на каждой входной комбинации дано в табл. 6.

Необходимо отметить, что для обеспечения самопроверяемости необходимо выполнение трех условий:

- формирование хотя бы по одному разу тестовых комбинаций для каждого из элементов преобразования ( $XOR$ ) – комбинаций из множества<sup>6</sup> {00, 01, 10, 11} [40];
- формирование хотя бы по одному разу каждого контрольного вектора  $WS(6, 2, 4)$  кода – векторов {00, 01, 10, 11};
- обнаружение в СВК всех ошибок, вызываемых неисправностями блоков  $F(x)$  и  $G(x)$ .

Из анализа табл. 6 следует, что условия 1 и 2 выполняются. Если бы они не выполнялись, можно было бы выбрать другие преобразуемые функции (это можно сделать в соответствии с формулой (2.1)  $8! = 40\,320$  способами без учета возможностей иного взвешивания разрядов информационного вектора).

Условие 3 можно проверить только при синтезе СВК на конкретной элементной базе по определенному методу (далее этого делать не будем, но при синтезе реального самопроверяемого устройства условие 3 должно быть выполнено).

Синтезируем самопроверяемое устройство и оценим показатели сложности его технической реализации. Будем использовать оценку в метрике числа входов логических элементов внутренней структуры устройства, которая имеет корреляцию с числом транзисторов, занимаемых устройством на кристалле [41]. При синтезе структур применим двухуровневую оптимизацию функций по методу Карно (также не будем использовать скобочных форм при сокращении записей булевых функций). На практике могут применяться и иные методы оптимизации функций и систем функций [42]. Однако в контексте настоящей статьи это не столь принципиально. Отметим, что могут использоваться и другие метрики, например непосредственно число транзисторов, необходимых для реализации устройства, или условный

<sup>6</sup> Для тестирования элемента  $XOR$  в канонической реализации [39] требуется подать на его входы все четыре входные комбинации. Если используется не каноническая реализация, то число тестовых комбинаций может быть уменьшено: тестовыми являются комбинации либо из подмножества {00, 01, 11}, либо из {00, 01, 10}, либо из {00, 10, 11}, либо из {01, 10, 11}. Но далее рассматривается именно каноническая реализация  $XOR$ , так как все остальные реализации будут ее частными случаями, а результаты исследований авторов легко применяются и в этих случаях.

показатель площади, занимаемой устройством на кристалле. Последний показатель применяется в известном интерпретаторе SIS, позволяющем автоматически выполнять процедуры оптимизации булевых функций и структур дискретных устройств, синтезированных в конкретных элементных базисах, и решать некоторые задачи их тестирования [43, 44].

Оптимизация функций, формируемых на выходах объекта диагностирования  $F(x)$ , дает следующий результат (сами процедуры оптимизации здесь не приведены ввиду их тривиальности):

$$\begin{aligned} f_8 &= \bar{x}_4 \bar{x}_3 \bar{x}_1 \vee \bar{x}_4 x_2 x_1 \vee x_4 x_2 \bar{x}_1 \vee x_4 x_3 \bar{x}_2 x_1; \\ f_7 &= x_4 \bar{x}_1 \vee \bar{x}_3 x_2 x_1 \vee x_3 \bar{x}_2 x_1 \vee \bar{x}_4 \bar{x}_2 x_1 \vee x_3 x_2 \bar{x}_1; \\ f_6 &= x_4 \bar{x}_2 x_1 \vee \bar{x}_4 x_2 \bar{x}_1 \vee \bar{x}_4 x_3 \bar{x}_1; \\ f_4 &= \bar{x}_3 \bar{x}_2 \bar{x}_1 \vee x_4 \bar{x}_3 x_2 x_1 \vee \bar{x}_4 \bar{x}_2 \bar{x}_1 \vee \bar{x}_4 x_3 x_2 x_1 \vee x_4 x_3 x_2 \bar{x}_1; \\ f_3 &= x_4 x_3 \bar{x}_2 x_1 \vee \bar{x}_4 x_2 \bar{x}_1 \vee \bar{x}_3 x_2 \bar{x}_1; \\ f_2 &= x_4 \bar{x}_2 \bar{x}_1 \vee \bar{x}_4 \bar{x}_3 \bar{x}_2 \vee x_3 x_1 \vee \bar{x}_4 x_3 x_2 \vee x_4 x_2 x_1; \\ f_1 &= x_3 \bar{x}_2 \bar{x}_1 \vee \bar{x}_4 \bar{x}_2 \bar{x}_1 \vee \bar{x}_4 \bar{x}_3 x_2 x_1 \vee x_4 x_3 x_2 x_1 \vee x_4 \bar{x}_3 \bar{x}_2 x_1 \vee x_4 \bar{x}_3 x_2 \bar{x}_1. \end{aligned}$$

Отсюда следует, что сложность реализации блока  $F(x)$  в метрике числа входов внутренних логических элементов определяется величиной  $L_{F(x)} = 146$  усл. ед. Оптимизация функций, реализуемых на выходах блока  $G(x)$ , приводит к следующему результату:

$$\begin{aligned} g_2 &= x_4 x_1 \vee x_3 \bar{x}_2 x_1 \vee x_4 \bar{x}_3 x_2; \\ g_1 &= x_3 x_1 \vee \bar{x}_4 x_1 \vee x_2 x_1 \vee x_4 \bar{x}_3 \bar{x}_1. \end{aligned}$$

Сложность реализации блока  $G(x)$  определяется величиной  $L_{G(x)} = 24$  усл. ед. Показатель сложности реализации устройства с СВК находится по формуле

$$L_{CED} = L_{F(x)} + L_{G(x)} + 2L_{XOR} + L_{G(f)} + L_{TRC} + 2L_{NOT}, \quad (3.1)$$

где  $L_{CED}$  – показатель сложности реализации устройства с СВК на основе выбранного кода Боуза–Лина<sup>7</sup>,  $L_{XOR} = 6$  усл. ед. – показатель сложности одного элемента  $XOR$ ,  $L_{NOT} = 1$  усл. ед. – показатель сложности инвертора в  $TSC$ ,  $L_{G(f)} = 32$  усл. ед. – показатель сложности кодера в  $TSC$ ,  $L_{TRC} = 12$  усл. ед. – показатель сложности  $TRC$  в  $TSC$ . Показатели для кодеров получаются с учетом того, что показатель сложности полного сумматора оценивается величиной  $L_{FA} = 14$  усл. ед., а полусумматора –  $L_{HA} = 6$  усл. ед. Данные величины можно найти непосредственно по внутренним структурам сумматора, полусумматора и модуля  $TRC$  [33].

Из выражения (3.1) имеем:

$$L_{CED} = 146 + 24 + 2 \cdot 6 + 32 + 12 + 2 \cdot 1 = 228 \text{ усл. ед.}$$

Для оценки полученной величины и сравнения используем структуру дублирования. Для нее показатель сложности реализации вычисляется по формуле [23]:

$$L_D = 2L_{F(x)} + nL_{NOT} + (n-1)L_{TRC}, \quad (3.2)$$

где  $n$  – число выходов устройства  $F(x)$ . Следуя за (3.2), запишем:

$$L_D = 2 \cdot 146 + 8 \cdot 1 + (8-1) \cdot 12 = 384 \text{ усл. ед.}$$

<sup>7</sup> Нижний индекс  $CED$  – от англ. concurrent error-detection.

<sup>8</sup> Два инвертора устанавливаются либо на выходах  $G(x)$ , либо на выходах  $G(f)$  для работы в парафазной логике элемента сравнения  $TRC$ .

**Таблица 7.** Сигналы на линиях СВК, реализованной на основе ЛКС с применением  $WS(5, 3, 8)$  кода с массивом весовых коэффициентов  $[1, 1, 2, 2, 2]$

$x_4$	$x_3$	$x_2$	$x_1$	$f_8$	$f_7$	$f_6$	$f_5$	$f_4$	$f_3$	$f_2$	$f_1$	$h_8$	$h_7$	$h_6$	$h_5$	$h_4$	$h_3$	$h_2$	$h_1$	$g_3$	$g_2$	$g_1$
0	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	0	0	0
0	0	0	1	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1
0	0	1	0	1	0	1	0	0	1	0	0	1	0	1	0	0	0	1	0	1	1	0
0	0	1	1	1	1	0	1	0	0	0	1	1	1	0	1	0	1	0	0	1	0	1
0	1	0	0	0	0	1	1	1	0	0	1	0	0	1	1	1	1	0	1	1	0	0
0	1	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	1	1
0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	1	0
0	1	1	1	1	1	0	0	0	1	0	1	0	1	0	0	1	0	1	1	0	0	1
1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0	1	1	0	0	1
1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	1	0	1	0
1	0	1	0	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	0	0	0	1
1	0	1	1	0	1	0	1	1	0	1	0	0	1	0	1	1	1	0	1	1	1	1
1	1	0	0	0	1	0	1	0	0	1	1	0	1	0	1	0	0	1	1	0	0	0
1	1	0	1	1	1	1	0	0	1	1	0	1	1	1	0	0	0	1	1	1	0	1
1	1	1	0	1	1	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	0	0
1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1

Оценим эффективность использования  $WS(6, 2, 4)$  кода для синтеза СВК на основе ЛКС по сравнению с применением дублирования:

$$\mu = \frac{L_{CED}}{L_D} \cdot 100\% \tag{3.3}$$

Вычислим показатель  $\mu$  для рассматриваемого примера:

$$\mu = \frac{228}{384} \cdot 100\% = 59.375\%$$

Таким образом, устройство, синтезированное на основе ЛКС с преобразованием только тех выходов объекта диагностирования, которые формируют проверочные символы в контрольном векторе  $WS(6, 2, 4)$  кода, оказывается более чем в 1.5 раза эффективнее по показателю сложности технической реализации, чем при дублировании.

3.2. Синтез схемы встроенного контроля на базе кода с модулем  $M = 8$ . Синтезируем СВК на основе ЛКС для примера из разд. 3.1 с применением другого кода Боуза–Лина –  $WS(5, 3, 8)$  кода с массивом весовых коэффициентов  $[w_8, w_7, w_6, w_5, w_4] = [1, 1, 2, 2, 2]$ . Структура организации СВК будет выглядеть так, как это представлено на рис. 7.

Отличие структуры рис. 7 от структуры рис. 5 состоит только лишь в том, что символ  $h_3$  здесь является проверочным для  $WS(5, 3, 8)$  кода и формируется с применением логической коррекции:  $h_3 = g_3 \oplus f_3$ . Также отличается тестер в структуре СВК: он включает в себя кодер  $WS(5, 3, 8)$  кода с показателем сложности  $L_{G(f)} = 32$  усл. ед., изображенный на рис. 8, и два модуля  $TRC$ .

Описание работы СВК дается в табл. 7, по которой синтезируется блок  $G(x)$ . Блок  $F(x)$  синтезирован так же, как и в предыдущем примере. При синтезе блока  $G(x)$  предварительно оптимизируем функции  $g_3, g_2$  и  $g_1$ :

$$g_3 = \bar{x}_4 \bar{x}_3 x_2 \vee \bar{x}_3 x_2 x_1 \vee \bar{x}_4 x_3 \bar{x}_2 \bar{x}_1 \vee x_4 x_3 \bar{x}_2 x_1 \vee x_4 x_3 x_2 \bar{x}_1;$$

$$g_2 = x_4 \bar{x}_3 x_1 \vee \bar{x}_4 x_2 \bar{x}_1 \vee x_4 x_2 x_1 \vee \bar{x}_4 x_3 \bar{x}_2 x_1;$$

$$g_1 = x_3 x_1 \vee \bar{x}_4 x_1 \vee x_2 x_1 \vee x_4 \bar{x}_3 \bar{x}_1.$$

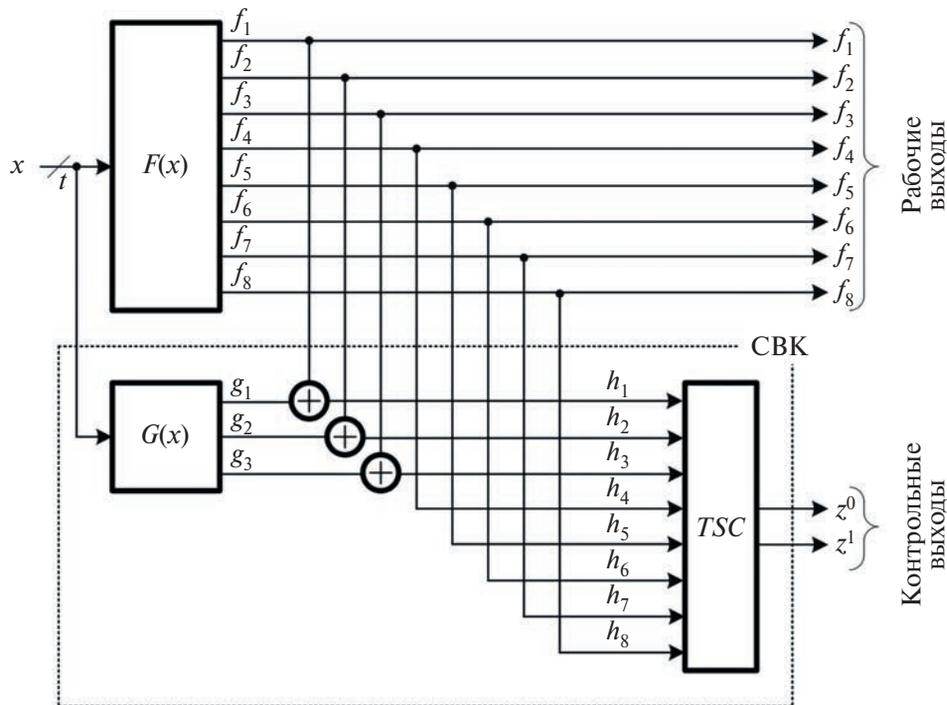


Рис. 7. Структура организации СВК на основе ЛКС для 8-выходного устройства с применением  $WS(5, 3, 8)$  кода

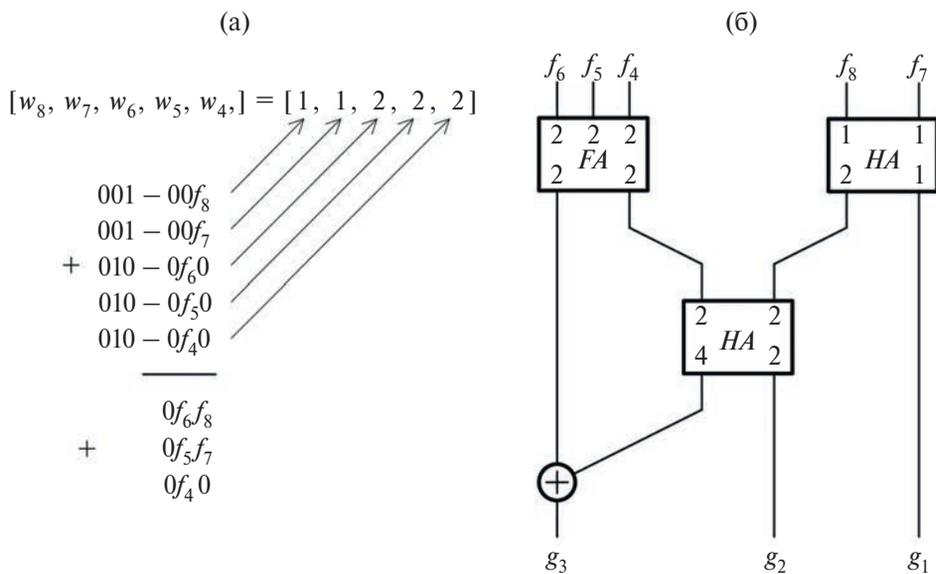


Рис. 8. Кодер  $WS(5, 3, 8)$  кода с массивом весовых коэффициентов  $[1, 1, 2, 2, 2]$ : а – схема построения, б – структура

Отсюда следует, что сложность реализации блока  $G(x)$  оценивается величиной  $L_{G(x)} = 53$  усл. ед.

Показатель сложности реализации устройства с СВК будет определяться как

$$L_{CED} = L_{F(x)} + L_{G(x)} + 3L_{XOR} + L_{G(f)} + 2L_{TRC} + 3L_{NOT}. \quad (3.4)$$

По формуле (3.4) имеем:

$$L_{CED} = 146 + 53 + 3 \cdot 6 + 32 + 2 \cdot 12 + 3 \cdot 1 = 276 \text{ усл. ед.}$$

**Таблица 8.** Сигналы на линиях СВК, реализованной на основе ЛКС с применением  $WS(5, 3, 8)$  кода с массивом весовых коэффициентов [2, 3, 4, 4, 2]

$x_4$	$x_3$	$x_2$	$x_1$	$f_8$	$f_7$	$f_6$	$f_5$	$f_4$	$f_3$	$f_2$	$f_1$	$h_8$	$h_7$	$h_6$	$h_5$	$h_4$	$h_3$	$h_2$	$h_1$	$g_3$	$g_2$	$g_1$
0	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	1	0	0	1	1	1
0	0	0	1	0	1	0	1	0	0	1	0	0	1	0	1	0	1	1	1	1	0	1
0	0	1	0	1	0	1	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1	0
0	0	1	1	1	1	0	1	0	0	0	1	1	1	0	1	0	0	0	1	0	0	0
0	1	0	0	0	0	1	1	1	0	0	1	0	0	1	1	1	0	1	0	0	1	1
0	1	0	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	1
0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	0	1	1	1	0	1
0	1	1	1	1	0	0	0	1	0	1	0	1	0	0	0	1	1	0	0	1	1	0
1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1
1	0	1	0	1	1	0	1	0	1	0	1	1	1	0	1	0	0	0	1	1	0	0
1	0	1	1	0	1	0	1	1	0	1	0	0	1	0	1	1	0	0	1	0	1	1
1	1	0	0	0	1	0	1	0	0	1	1	0	1	0	1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	0	0	1	1	0	1	1	1	0	0	0	0	1	1	1	1
1	1	1	0	1	1	0	0	1	0	0	0	1	1	0	0	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1

Используя (3.3), сравним полученный показатель для структуры на основе ЛКС с дублированием:

$$\mu = \frac{276}{384} \cdot 100\% = 71.875\%.$$

Устройство, синтезированное на базе ЛКС с преобразованием только тех выходов объекта диагностирования, которые формируют проверочные символы в контрольном векторе  $WS(5, 3, 8)$  кода, также оказывается эффективнее по показателям сложности технической реализации, чем при использовании дублирования. Однако, как показывает анализ табл. 7, для тестера не формируются две комбинации –  $\langle 010 \rangle$  и  $\langle 111 \rangle$ , что не дает возможности реализации полностью самопроверяемой СВК. В данном случае можно применить тестер, реализуемый на высоконадежных элементах, как “последний сторож”, и обеспечивать обнаружение неисправностей в объекте диагностирования и в других составляющих СВК [5].

Еще одним вариантом обеспечения самопроверяемости является подбор других весовых коэффициентов. В соответствии с формулой (1.4) можно взвесить разряды информационного вектора  $WS(5, 3, 8)$  кода следующим числом способов:  $N_{WS(5,3)} = C_{5+6}^3 = 462$ . И это без учета перестановок выходов! Исследования показывают, что самопроверяемая СВК дает  $WS(5, 3, 8)$  код с массивом весовых коэффициентов  $[w_8, w_7, w_6, w_5, w_4] = [2, 3, 4, 4, 2]$ .

В табл. 8 представлены сигналы на линиях СВК, реализованной на основе ЛКС с применением  $WS(5, 3, 8)$  кода с массивом весовых коэффициентов  $[w_8, w_7, w_6, w_5, w_4] = [2, 3, 4, 4, 2]$ . Кодер данного кода изображен на рис. 9. Показатель сложности его реализации равен  $L_{G(f)} = 32$  усл. ед. Сложность реализации блока  $G(x)$ , полученного аналогичным предыдущему способом, определена величиной  $L_{G(x)} = 56$  усл. ед., поскольку он описывается функциями:

$$g_3 = x_4 \bar{x}_1 \vee x_4 x_3 \bar{x}_2 \vee \bar{x}_4 \bar{x}_3 \bar{x}_2 \vee \bar{x}_2 x_3 x_2;$$

$$g_2 = \bar{x}_4 \bar{x}_2 \bar{x}_1 \vee \bar{x}_3 \bar{x}_2 \bar{x}_1 \vee \bar{x}_4 \bar{x}_3 \bar{x}_1 \vee x_4 x_3 x_1 \vee x_4 x_3 x_2 \vee x_3 x_2 x_1 \vee x_4 x_3 x_1;$$

$$g_1 = \bar{x}_4 \bar{x}_2 \vee x_4 x_1 \vee \bar{x}_3 \bar{x}_2 \vee x_3 x_2 \bar{x}_1.$$

Пользуясь (3.4), находим:

$$L_{CED} = 146 + 56 + 3 \cdot 6 + 32 + 2 \cdot 12 + 3 \cdot 1 = 279 \text{ усл. ед.}$$

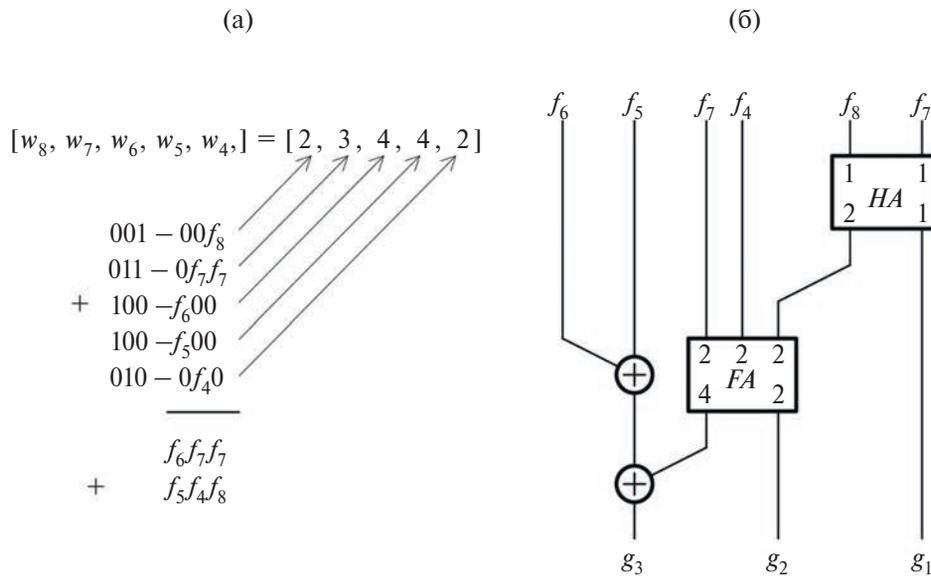


Рис. 9. Кодер  $WS(5, 3, 8)$  кода с массивом весовых коэффициентов  $[2, 3, 4, 4, 2]$ :  
 а – схема построения, б – структура

С применением (3.3) оценим полученный показатель для сравнения с показателем сложности реализации структуры на основе дублирования:

$$\mu = \frac{279}{384} \cdot 100\% = 72.656\%.$$

Отсюда следует, что определен вариант реализации самопроверяемого устройства с показателями структурной избыточности, соизмеримыми с предыдущим несопроверяемым вариантом реализации.

Возможность выбора способа взвешивания позволяет на практике добиваться самопроверяемости СВК, а приведенные в данном разделе статьи примеры демонстрируют эффективность использования взвешенных кодов Боуза–Лина при синтезе самопроверяемых устройств.

**Заключение.** Взвешенные коды Боуза–Лина представляют собой обобщенный класс кодов с суммированием с простыми схемами кодеров, что позволяет их эффективно применять при синтезе СВК. В статье установлено, что число способов взвешивания для построения помехозащищенного кода определяется величинами  $N_{WS(m, 2, 4)} = C_{m+2}^m$  и  $N_{WS(m, 3, 8)} = C_{m+6}^m$ . Таким образом, для заданного  $m$  может быть построено большое количество кодов, обладающих различными характеристиками обнаружения ошибок и разными по сложности схемами кодеров. Использование ЛКС для синтеза СВК позволяет построение огромного числа вариантов самопроверяемых устройств и выбора среди них наилучшего по характеристикам. Исследования показывают, что такое количество способов построения устройств в большинстве случаев дает возможность синтеза полностью самопроверяемых структур даже в тех случаях, когда этого не удается добиться использованием типовой структуры на основе дублирования.

Установленные в статье особенности, связанные с изменением числа способов построения СВК на базе ЛКС при помощи взвешенных кодов Боуза–Лина с различными значениями  $M$ , могут учитываться при выборе способа реализации самопроверяемых цифровых устройств. Результаты, полученные в работе, могут быть обобщены для случаев произвольных значений модуля, в том числе тех, которые дают наиболее просто обеспечивать самопроверяемость СВК, – модулей

$$M \in \left\{ 2^1, 2^2, \dots, 2^{\left\lfloor \log_2 \left( 1 + \sum_{i=1}^m w_i \right) \right\rfloor - 1} \right\}.$$

Дальнейшие исследования могут быть связаны с изучением различных алгоритмов синтеза СВК на основе взвешенных кодов Боуза–Лина, с постановкой большого числа экспериментов с тестовыми схемами [45, 46], а также с поиском способов взвешивания разрядов, исходя из известных статистических данных о формировании единичных и нулевых значений на каждом из выходов объекта диагностирования для обеспечения самопроверяемости структур СВК. Кроме того, важными для практики являются исследования возможностей применения разрабатываемых способов построения СВК для реализации самопроверяемых структур на различной элементной базе, в том числе на современных программируемых устройствах.

Использование ЛКС совместно с взвешенными кодами Боуза–Лина представляется перспективным направлением в вопросах реализации устройств и систем с обнаружением неисправностей.

### СПИСОК ЛИТЕРАТУРЫ

1. *Пархоменко П.П., Согомонян Е.С.* Основы технической диагностики (оптимизация алгоритмов диагностирования, аппаратные средства). М.: Энергоатомиздат, 1981, 320 с.
2. *Дрозд А.В., Харченко В.С., Антошук С.Г., Дрозд Ю.В., Дрозд М.А., Сулима Ю.Ю.* Рабочее диагностирование безопасных информационно-управляющих систем / Под ред. А.В. Дрозда и В.С. Харченко. Харьков: Национальный аэрокосмический ун-т им. Н.Е. Жуковского “ХАИ”, 2012. 614 с.
3. *Kharchenko V., Kondratenko Yu., Kasprzyk J.* Green IT Engineering: Concepts, Models, Complex Systems Architectures // Springer Book Series “Studies in Systems, Decision and Control”. 2017. V. 74. 305 p.; <https://doi.org/10.1007/978-3-319-44162-7>
4. *Mikoni S.* Top Level Diagnostic Models of Complex Objects // Lecture Notes in Networks and Systems. 2022. V. 442. P. 238–249; [https://doi.org/10.1007/978-3-030-98832-6\\_21](https://doi.org/10.1007/978-3-030-98832-6_21)
5. *Гавзов Д.В., Сапожников В.В., Сапожников Вл.В.* Методы обеспечения безопасности дискретных систем // АИТ. 1994. № 8. С. 3–50.
6. *Согомонян Е.С., Слабаков Е.В.* Самопроверяемые устройства и отказоустойчивые системы. М.: Радио и связь, 1989. 208 с.
7. *Göessel M., Ocheretny V., Sogomonyan E., Marienfeld D.* New Methods of Concurrent Checking: Edition 1. Dordrecht: Springer Science+Business Media B.V., 2008. 184 p.
8. *Goessel M., Graf S.* Error Detection Circuits. London: McGraw-Hill, 1994. 261 p.
9. *Аксёнова Г.П., Согомонян Е.С.* Построение самопроверяемых схем встроенного контроля для автоматов с памятью // АИТ. 1975. № 7. С. 132–142.
10. *Гессель М., Морозов А.В., Сапожников В.В., Сапожников Вл.В.* Контроль комбинационных схем методом логического дополнения // АИТ. 2005. № 8. С. 161–172.
11. *Гессель М., Согомонян Е.С.* Построение самотестируемых и самопроверяемых комбинационных устройств со слабозависимыми выходами // АИТ. 1992. № 8. С. 150–160.
12. *Sogomonyan E.S., Gössel M.* Design of Self-Testing and On-Line Fault Detection Combinational Circuits with Weakly Independent Outputs // J. Electronic Testing: Theory and Applications. 1993. V. 4. Iss. 4. P. 267–281; <https://doi.org/10.1007/BF00971975>
13. *Saposhnikov V.I., Dmitriev A., Goessel M., Saposhnikov V.V.* Self-Dual Parity Checking – A New Method for on Line Testing // Proc. 14<sup>th</sup> IEEE VLSI Test Sympos. USA, Princeton, 1996. P. 162–168.
14. *Efanov D.V., Pivovarov D.V.* The Hybrid Structure of a Self-Dual Built-In Control Circuit for Combinational Devices with Pre-Compression of Signals and Checking of Calculations by Two Diagnostic Parameters // Proc. 19th IEEE East-West Design & Test Symposium (EWDTS’2021). Batumi, Georgia, 2021. P. 200–206; <https://doi.org/10.1109/EWDTS52692.2021.9581019>
15. *Ефанов Д.В., Погодина Т.С.* Исследование свойств самодвойственных комбинационных устройств с контролем вычислений на основе кодов Хэмминга // Информатика и автоматизация. 2023. Т. 22. № 2. С. 349–392; <https://doi.org/10.15622/ia.22.2.5>
16. *Гессель М., Морозов А.В., Сапожников В.В., Сапожников Вл.В.* Логическое дополнение новый метод контроля комбинационных схем // АИТ. 2003. № 1. С. 167–176.
17. *Sen S.K.* A Self-Checking Circuit for Concurrent Checking by 1-out-of-4 code with Design Optimization using Constraint Don’t Cares // National Conf. on Emerging Trends and Advances in Electrical Engineering and Renewable Energy (NCEEERE 2010). Sikkim: Sikkim Manipal Institute of Technology, 2010.
18. *Das D.K., Roy S.S., Dmitriev A., Morozov A., Gössel M.* Constraint Don’t Cares for Optimizing Designs for Concurrent Checking by 1-out-of-3 Codes // Proc. 10th Intern. Workshops on Boolean Problems. Freiberg, Germany, 2012. P. 33–40.
19. *Пивоваров Д.В.* Построение систем функционального контроля многовыходных комбинационных схем методом логического дополнения по равновесным кодам // Автоматика на транспорте. 2018. Т. 4. № 1. С. 131–149.
20. *Efanov D.V., Sapozhnikov V.V., Sapozhnikov V.I., Pivovarov D.V.* Synthesis of Built-in Self-Test Control Circuits Based on the Method of Boolean Complement to Constant-Weight 1-out-of-n Codes // Automatic Control and Computer Sciences. 2019. V. 53. Iss. 6. P. 481–491; <https://doi.org/10.3103/S014641161906004X>

21. *Morozov M., Saposhnikov V.V., Saposhnikov V.V., Goessel M.* New Self-Checking Circuits by Use of Berger-codes // Proc. IEEE Intern. On-Line Testing Workshop. Palma De Mallorca, Spain, 2000. P. 171–176.
22. *Berger J.M.* A Note on Error Detection Codes for Asymmetric Channels // Information and Control. 1961. V. 4. Iss. 1. P. 68–73;  
[https://doi.org/10.1016/S0019-9958\(61\)80037-5](https://doi.org/10.1016/S0019-9958(61)80037-5)
23. *Efanov D.V., Sapozhnikov V.V., Sapozhnikov V.V.* The Self-Checking Concurrent Error-Detection Systems Synthesis Based on the Boolean Complement to the Bose-Lin Codes with the Modulo Value  $M=4$  // Electronic Modeling. 2021. V. 43. Iss. 1. P. 28–45;  
<https://doi.org/10.15407/emodel.43.01.028>
24. *Слабаков Е.В.* Построение полностью самопроверяемых комбинационных устройств с использованием остаточных кодов // АИТ. 1979. № 10. С. 133–141.
25. *Слабаков Е.В., Согомонян Е.С.* Самопроверяемые вычислительные устройства и системы (обзор) // АИТ. 1981. № 11. С. 147–167.
26. *Piestrak S.J.* Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995. 111 p.
27. *Das D., Touba N.A.* Synthesis of Circuits with Low-Cost Concurrent Error Detection Based on Bose-Lin Codes // J. Electronic Testing: Theory and Applications. 1999. V. 15. Iss. 1–2. P. 145–155;  
<https://doi.org/10.1023/A:1008344603814>
28. *Ефанов Д.В., Сапожников В.В., Сапожников В.В.* Применение модульных кодов с суммированием для построения систем функционального контроля комбинационных логических схем // АИТ. 2015. № 10. С. 152–169.
29. *Сапожников В.В., Сапожников В.В., Ефанов Д.В., Черепанова М.Р.* Модульные коды с суммированием в системах функционального контроля. I. Свойства обнаружения ошибок кодами в информационных векторах // Электронное моделирование. 2016. Т. 38. № 2. С. 27–48.
30. *Сапожников В.В., Сапожников В.В., Ефанов Д.В., Черепанова М.Р.* Модульные коды с суммированием в системах функционального контроля. II. Уменьшение структурной избыточности систем функционального контроля // Электронное моделирование. 2016. Т. 38. № 3. С. 47–61.
31. *Ефанов Д.В., Осадчий Г.В., Зуева М.В.* Особенности обнаружения ошибок кодами с суммированием единичных информационных разрядов в кольце вычетов по заданному модулю в схемах встроенного контроля, синтезированных на основе метода логического дополнения // Автоматика на транспорте. 2021. Т. 7. № 2. С. 284–314;  
<https://doi.org/10.20295/2412-9186-2021-7-2-284-314>
32. *Efanov D., Osadchy G., Zueva M.* Specifics of Error Detection with Modular Sum Codes in Concurrent Error-Detection Circuits Based on Boolean Complement Method // Proc. IEEE East-West Design & Test Sympos. (EWDTS'2021). Batumi, Georgia, 2021. P. 59–69;  
<https://doi.org/10.1109/EWDTS52692.2021.9581036>
33. *Сапожников В.В., Сапожников В.В., Ефанов Д.В.* Коды с суммированием для систем технического диагностирования. Т. 1. Классические коды Бергера и их модификации. М.: Наука, 2020. 383 с.
34. *Сапожников В.В., Сапожников В.В., Ефанов Д.В.* Коды с суммированием для систем технического диагностирования. Т. 2. Взвешенные коды с суммированием. М.: Наука, 2021. 455 с.
35. The On-Line Encyclopedia of Integer Sequences электронный ресурс (режим доступа: <https://oeis.org/>, 14.04.2024 г.).
36. *Грэхем Р.Л., Кнут Д.Э., Паташник О.* Конкретная математика. Математические основы информатики, 2-е изд. / Пер. с англ. М.: ООО “И.Д. Вильямс”, 2017. 784 с.
37. *Ефанов Д.В.* Синтез самопроверяемых комбинационных устройств на основе метода логической коррекции сигналов с применением кодов Боуза–Лина // Информационные технологии. 2023. Т. 29. № 10. С. 503–511;  
<https://doi.org/10.17587/it.29.503-511>
38. *Carter W.C., Duke K.A., Schneider P.R.* Self-Checking Error Checker for Two-Rail Coded Data // United States Patent Office, Filed July 25, 1968. Ser. 747533, patented Jan. 26. N. Y., 1971, 10 p.
39. *Harris D.M., Harris S.L.* Digital Design and Computer Architecture. Morgan Kaufmann, 2012. 569 p.
40. *Аксюнова Г.П.* Необходимые и достаточные условия построения полностью проверяемых схем свертки по модулю 2 // АИТ. 1979. № 9. С. 126–135.
41. *Сапожников В.В., Сапожников В.В.* Самопроверяемые дискретные устройства. СПб.: Энергоатомиздат, 1992. 224 с.
42. *Закревский А.Д., Поттосин Ю.В., Черемисинова Л.Д.* Логические основы проектирования дискретных устройств. М.: Физматлит, 2007. 592 с.
43. *Sentovich E.M., Singh K.J., Moon C., Savoj H., Brayton R.K., Sangiovanni-Vincentelli A.* Sequential Circuit Design Using Synthesis and Optimization // Proc. IEEE Intern. Conf. on Computer Design: VLSI in Computers & Processors. Cambridge, MA, USA, 1992. P. 328–333;  
<https://doi.org/10.1109/ICCD.1992.276282>
44. *Sentovich E.M., Singh K.J., Lavagno L., Moon C., Murgai R., Saldanha A. et al.* SIS: A System for Sequential Circuit Synthesis. Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, Berkeley: University of California, 1992, 45 p.
45. Collection of Digital Design Benchmarks электронный ресурс (режим доступа: <https://ddd.fit.cvut.cz/www/prj/Benchmarks/>, 14.04.2024 г.).
46. *McElvain K.* IWLS'93 Benchmark Set: Version 4.0. Distributed as a Part of IWLS'93 Benchmark Set, 1993.