



УДК 681.518.5:004.052.32

© 2018 г. **Д.В. Ефанов**, д-р техн. наук  
(Российский университет транспорта (МИИТ)),

**Т. Тэн**

(Цзянсуский педагогический университет)

## СПОСОБ СИНТЕЗА ГЕНЕРАТОРОВ КОДОВ С СУММИРОВАНИЕМ ВЗВЕШЕННЫХ ИНФОРМАЦИОННЫХ РАЗРЯДОВ

Предложен способ синтеза генераторов кодов с суммированием взвешенных информационных разрядов, основанный на разложении весовых коэффициентов по степеням числа 2 и использовании стандартных схем сумматоров и полусумматоров. Метод позволяет синтезировать простые структуры генераторов кодов с суммированием. Разработан подход к применению предложенного метода для использования в качестве основы мультиплексоров, входящих в структуры логических ячеек современных программируемых логических интегральных схем. Показаны возможности применения разработанного способа синтеза генераторов для построения структур модульных и модифицированных взвешенных кодов с суммированием.

**Ключевые слова:** код с суммированием взвешенных информационных разрядов, генератор, полный сумматор, полусумматор, мультиплексор.

DOI: 10.22250/isu.2018.58.82-91

### Введение

Взвешенные коды с суммированием представляют собой большой класс делимых кодов, ориентированных на обнаружение ошибок в информационных векторах [1]. Такие коды широко используются при построении систем диагностирования устройств автоматики и вычислительной техники [2 – 6].

Для построения взвешенных кодов с суммированием может быть использовано два подхода. Первый основан на приписывании весовых коэффициентов разрядам информационного вектора и дальнейшем суммировании весовых коэффициентов единичных информационных разрядов для получения суммарного веса информационного вектора. Такой подход предложен в [7]. Второй подход основан на приписывании весовых коэффициентов переходам между разрядами, за-

нимающими соседние позиции в информационном векторе, и последующем суммировании весовых коэффициентов разрядов тех переходов, значения разрядов между которыми различны. Этот способ построения кода впервые описан в [8]. Получаемые по первому и второму подходам коды обозначают как  $WS(m,k)$  и  $WT(m,k)$  коды, отдельно указывая последовательность весовых коэффициентов разрядов или переходов.

Принципы построения  $WS(m,k)$ - и  $WT(m,k)$ -кодов похожи. Однако сами свойства кодов существенно различаются. Например,  $WS(m,k)$ -коды с любыми последовательностями весовых коэффициентов обнаруживают все монотонные (однонаправленные) искажения в информационных векторах, тогда как  $WT(m,k)$ -кодам данная особенность не свойственна [1].

Генераторы (кодеры) взвешенных кодов обоих типов имеют похожие структуры. Со схемотехнической точки зрения вообще можно считать, что  $WT(m,k)$ -коды представляют собой некоторое обобщение  $WS(m,k)$ -кодов. Вообще, генератор любого  $WT(m,k)$ -кода может быть получен непосредственно из генератора  $WS(m-1,k)$ -кода путем добавления каскада сумматоров по модулю два, реализующего функции активации переходов [9]. Таким образом, интерес представляет вопрос синтеза генераторов взвешенных кодов с суммированием. В данной работе рассмотрено решение этой задачи с применением стандартной элементной базы в виде полусумматоров и полных сумматоров [10, 11]. Кроме того, представлен способ расширения действия алгоритма на применение мультиплексоров – устройств, входящих в структуры современных программируемых логических интегральных схем [12].

### **Постановка задачи**

Требуется разработать универсальный способ построения генераторов взвешенных кодов с суммированием, который позволит учитывать особенности весовых коэффициентов разрядов, а также будет легко адаптироваться к использованию при синтезе устройств автоматики на элементах программируемой логики. При этом в качестве основных структурных единиц для синтеза генератора предлагается выбор стандартных схем сумматоров одноразрядных двоичных чисел.

### **Синтез генераторов взвешенных кодов с суммированием**

Выбор в качестве элементной базы полусумматоров и полных сумматоров обусловлен использованием операций суммирования при построении кодов. Сумматоры осуществляют сложение одноразрядных двоичных чисел, поступающих на их входы. Полусумматор (full adder, FA) имеет два входа, а полный сум-

матор (half adder, HA) – три входа. В обоих вариантах устройств по два выхода – суммирования (sum, S) и переноса (carry, C). Полусумматоры и сумматоры имеют стандартные реализации и присутствуют в любых средствах автоматизированного проектирования логических устройств [12]. Данная элементная база широко применяется в задачах синтеза контрольных схем для различных избыточных кодов [13, 14].

Генераторы взвешенных кодов на полусумматорах и полных сумматорах могут строиться различными способами [15]. Например, каждый весовой коэффициент последовательности может быть представлен в виде двоичного числа, а сам генератор может быть напрямую построен в виде сумматора этих двоичных чисел. Такой подход даст наиболее сложные структуры генераторов взвешенных кодов. Другой подход подразумевает представление значений весовых коэффициентов в виде двоичных чисел и группировку их по возможности суммирования без использования операции переноса, а затем суммирование полученных таким образом двоичных чисел. Данный способ позволяет несколько упростить структуру генератора [16]. В ходе исследований установлено, что для синтеза генераторов взвешенных кодов наиболее целесообразно адаптировать подход, описанный в [17] для синтеза генераторов классических кодов Бергера [7].

Синтез генератора  $WS(m,k)$ -кода осуществляется следующим образом:

1. Весовые коэффициенты раскладываются на суммы степеней числа 2.
2. Определяется количество чисел  $i$ -й степени числа 2 в разложении числа  $N_i$ .
3. Полагается  $i=0$ .
4. Синтезируется  $i$ -й каскад генератора, содержащий  $\frac{N_i-1}{2} \pmod{2}$  полусумматоров и  $\left\lfloor \frac{N_i-1}{2} \right\rfloor$  полных сумматоров.
5.  $i = i+1$ .
6. Проверяется условие  $i=i_{\max}$ ? Если да, то генератор построен, если нет – то реализуется следующий шаг.
7. Определяется число выходов переноса каждого сумматора  $i-1$  каскада – число  $N_{C_{i-1}}$ .
8. Корректируется число  $N_i$ :  $N_i = N_i + N_{C_{i-1}}$ .
9. Повторяются операции 4 – 6.

Для примера на рис. 1 реализована структурная схема генератора  $WS(6,4)$ -кода с последовательностью весовых коэффициентов

$$[w_6, w_5, w_4, w_3, w_2, w_1] = [1, 1, 4, 3, 1, 5].$$

Также показаны сигналы на всех линиях схемы генератора при поступлении на входы информационного вектора  $\langle f_6 f_5 f_4 f_3 f_2 f_1 \rangle = \langle 011001 \rangle$ .

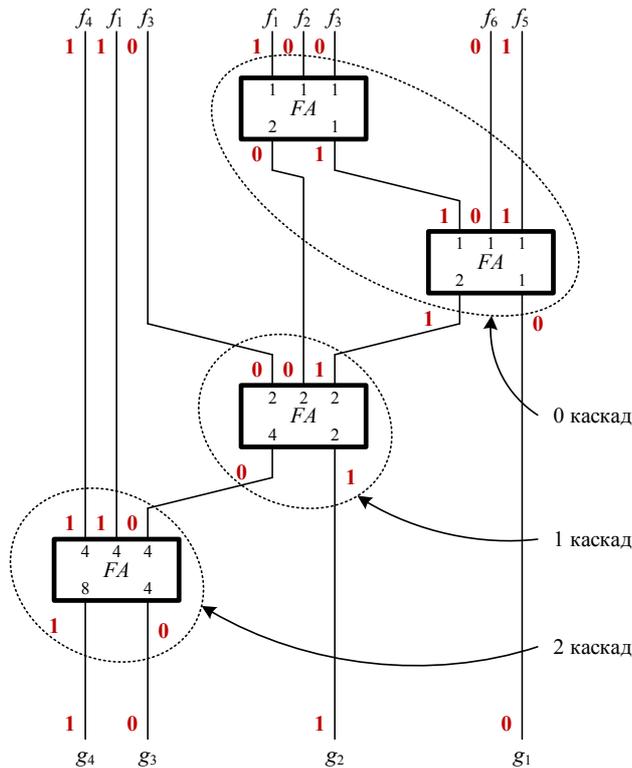


Рис. 1. Генератор  $WS(6,4)$ -кода с последовательностью весовых коэффициентов  $[1,1,4,3,1,5]$ , реализованный на стандартных сумматорах.

Для построения генератора  $WS(m,k)$ -кода по представленному выше алгоритму выполнено следующее разложение весовых коэффициентов:  $w_1 = 2^2 + 2^0$ ;  $w_2 = w_3 = w_6 = 2^0$ ;  $w_3 = 2^1 + 2^0$ ;  $w_4 = 2^2$ . Количество различных степеней числа 2 равно:  $N_0 = 5$ ;  $N_1 = 1$ ;  $N_2 = 2$ . Для нулевого каскада генератора, осуществляющего суммирование разрядов с весами  $2^0$ , потребовалось  $\left\lfloor \frac{5-1}{2} \right\rfloor = 2$  полных сумматора и  $\left(\frac{5-1}{2}\right) \pmod{2} = 0$  полусумматоров. Всего –  $\left\lfloor \frac{5}{2} \right\rfloor = 2$  сумматора. При построении первого каскада генератора к числу  $N_1 = 1$  следует прибавить число выходов переноса каждого из сумматоров первого каскада – число  $N_{C_0} = 2$ . Далее суммируются разряды с весами  $2^1$ , для чего требуются  $\left\lfloor \frac{(1+2)-1}{2} \right\rfloor = 1$  полный сумматор и  $\left(\frac{(1+2)-1}{2}\right) \pmod{2} = 0$  полусумматоров. Второй каскад генератора образован одним полным сумматором, так как числа  $N_2 = 2$  и  $N_{C_1} = 1$  (число полных сумматоров равно  $\left\lfloor \frac{(2+1)-1}{2} \right\rfloor = 1$ , а полусумматоров –  $\left(\frac{(2+1)-1}{2}\right) \pmod{2} = 0$ ).

Отметим, что при разложении весовых коэффициентов наиболее эффективным является разложение по максимальным степеням числа 2. Например, число 5 следует раскладывать так:  $2^2 + 2^0$  (а не так:  $2^1 + 2^1 + 2^0$ ).

Для оценки сложности полученной структуры может быть использовано количество стандартных функциональных элементов или количество входов внутренних логических элементов. Наиболее удобной является оценка сложности реализации устройств в показателях сложности библиотеки стандартных функциональных элементов *stdcell2\_2.genlib* [18], что позволяет получить единственное относительное число, напрямую связанное с размерами занимаемой устройством площади на кристалле. В рассматриваемой библиотеке элементов сложность реализации полусумматора оценивается величиной  $L_{HA} = 72$  (двухвходовый элемент *XOR* и двухвходовый элемент *AND*), а полного сумматора – величиной  $L_{FA} = 176$  (два полусумматора и двухвходовый элемент *OR*). С учетом этих данных сложность технической реализации генератора *WS(6,4)*-кода с последовательностью весовых коэффициентов [1,1,4,3,1,5] определяется величиной  $L_{WS} = 704$ .

На рис. 2 представлены оценки сложности технической реализации генераторов всех *WS(6,4)*-кодов с числом контрольных разрядов  $k = \lceil \log_2(m+1) \rceil$ .

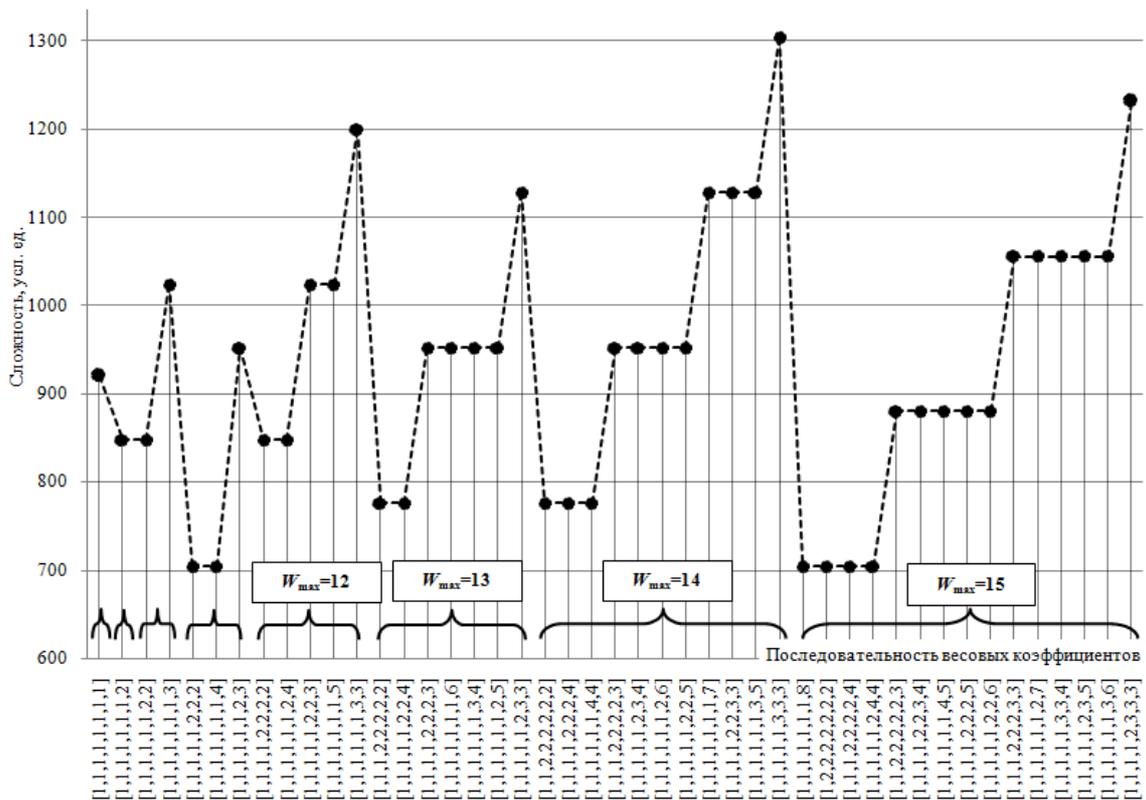


Рис. 2. Сложность технической реализации генераторов *WS(8,4)*-кодов.

При этом, коды расположены в последовательности возрастания числа  $W_{max}$ , что позволяет судить о закономерностях в изменениях показателей сложности реализации. Следует отметить тот факт, что для 15 из 44 взвешенных кодов с суммированием (за исключением классического кода с суммированием) сложность реализации получается даже меньшей, чем сложность реализации генератора *S(8,4)*-кода по этому же методу. Кроме того, как показано выше, любым

взвешенным кодом обнаруживается гораздо большее количество ошибок, чем  $S(m,k)$ -кодом. С увеличением значения  $W_{\max}$  число не обнаруживаемых ошибок уменьшается. При этом для каждого значения  $W_{\max}$  может быть выбрана последовательность весовых коэффициентов, дающая меньшую по сравнению с  $S(m,k)$ -кодом сложность реализации.

Для реализации генераторов  $WS(m,k)$ -кодов может быть использована и другая элементная база, – например, мультиплексоры [19]. Реализация устройств на мультиплексорах перспективна, поскольку данные устройства входят в состав логических блоков современных программируемых логических интегральных схем. На рис. 3 представлены наиболее простые варианты реализации сумматоров на мультиплексорах с одним и двумя адресными входами.

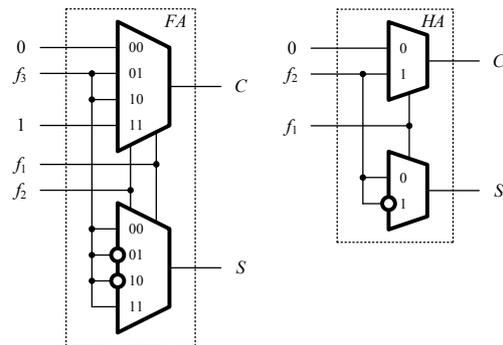


Рис. 3. Реализация сумматоров на мультиплексорах.

Для реализации генератора взвешенного кода на мультиплексорах требуется сначала построить его на сумматорах по предложенному выше алгоритму, а затем осуществить замену каждого сумматора на стандартную его реализацию на мультиплексорах. На рис. 4 представлена реализация генератора  $WS(6,4)$ -кода с последовательностью весовых коэффициентов  $[1,1,4,3,1,5]$  на мультиплексорах.

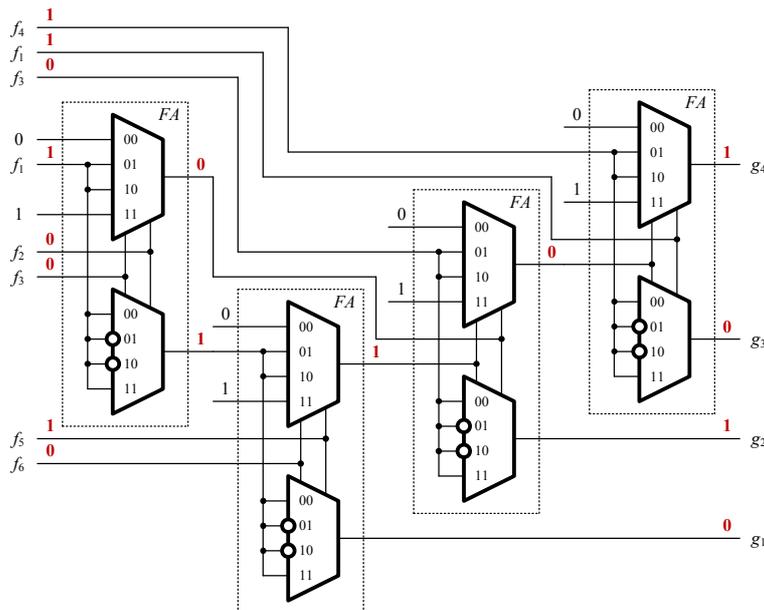


Рис. 4. Генератор  $WS(6,4)$ -кода с последовательностью весовых коэффициентов  $[1,1,4,3,1,5]$ ,

реализованный на мультиплексорах.

### Синтез генераторов модифицированных взвешенных кодов

Описанный здесь способ построения генераторов может быть использован не только для синтеза генераторов  $WS(m,k)$ - и  $WT(m,k)$ -кодов. Данные коды являются «базовыми» для построения других модифицированных кодов с суммированием: модульно взвешенных кодов с суммированием [5, 20] и модифицированных кодов с суммированием [21, 22].

Структурная схема генераторов модульно взвешенных кодов с суммированием приведена на рис. 5.

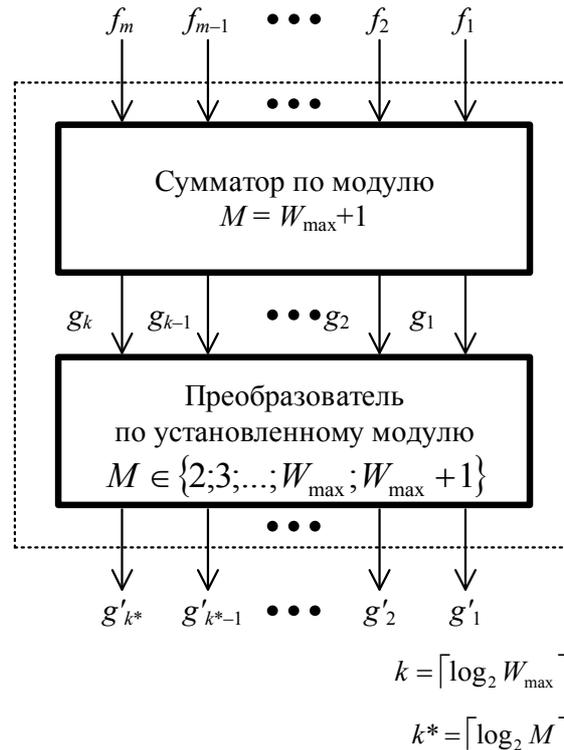


Рис. 5. Структурная схема генераторов модульно взвешенных кодов с суммированием.

Генератор синтезируется в виде устройства, включающего два блока. Первый блок представляет собой генератор «базового»  $WS(m,k)$ - или  $WT(m,k)$ -кода и является сумматором весовых коэффициентов единичных информационных разрядов по модулю  $M = W_{\max} + 1$ . Число выходов сумматора определяется числом разрядов двоичного числа, представляющего собой максимальное значение веса (число  $W_{\max} = w_1 + w_2 + \dots + w_{m-1} + w_m$ ):  $k = \lceil \log_2 W_{\max} \rceil$ , где запись  $\lceil \dots \rceil$  обозначает целое сверху от вычисляемого значения. Второй блок является устройством преобразования двоичного числа  $\langle g_k g_{k-1} \dots g_2 g_1 \rangle$  в двоичное число  $\langle g'_{k^*} g'_{k^*-1} \dots g'_2 g'_1 \rangle$ , представляющего наименьший неотрицательный вычет числа  $\langle g_k g_{k-1} \dots g_2 g_1 \rangle$  (или  $W_{\max}$  в десятичном эквиваленте) по заранее выбранному модулю  $M \in \{2; 3; \dots; W_{\max} - 1; W_{\max}\}$ . Таким образом, количество разрядов данного числа определяется величиной  $k^* = \lceil \log_2 M \rceil$ . Преобразователь может быть синтезирован ме-

тодами, изложенными в [23, 24].

Наиболее просто строятся генераторы модульно взвешенных кодов с суммированием со значениями модулей  $M \in \{2; 4; 8; \dots; 2^{\lceil \log_2 W_{\max} \rceil - 2}; 2^{\lceil \log_2 W_{\max} \rceil - 1}\}$ . Для таких кодов преобразователь в структурной схеме на рис. 5 как таковой отсутствует, а на этапе синтеза генератора после построения сумматора по модулю  $M = W_{\max} + 1$  производится упрощение его структуры путем удаления неиспользуемых выходов с последующей заменой неиспользуемых полусумматоров и полных сумматоров на сумматоры по модулю два. Например, на рис. 6 представлен генератор  $WS(6,3)$ -кода с последовательностью весовых коэффициентов  $[1, 1, 4, 3, 1, 5]$  и модулем  $M = 8$ .

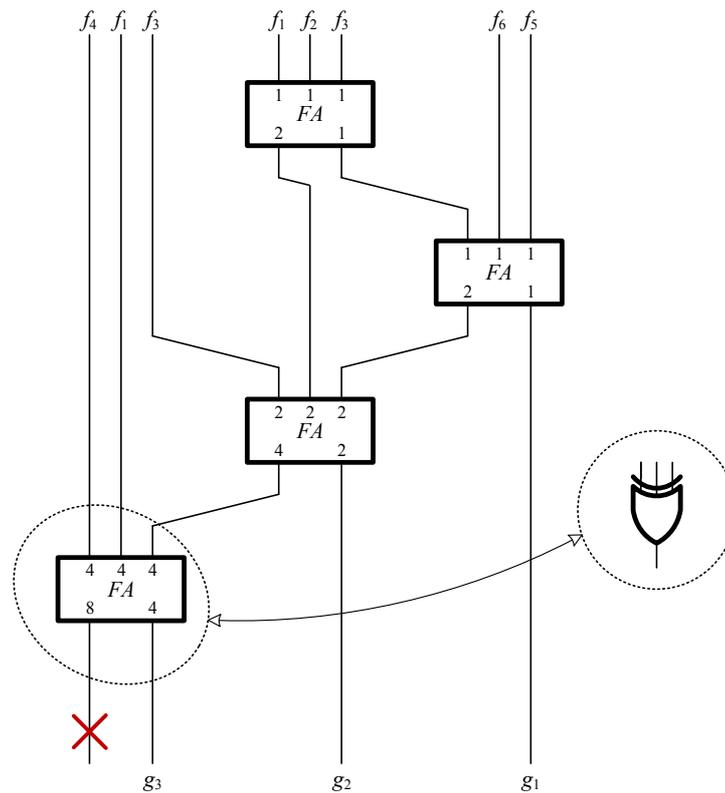


Рис. 6. Генератор  $WS(6,3)$ -кода с последовательностью весовых коэффициентов  $[1, 1, 4, 3, 1, 5]$  и модулем  $M=8$ , реализованный на стандартных сумматорах.

Модульные коды являются базовыми и для всех остальных модификаций кодов с суммированием [21], по этой причине представленный в данной работе метод можно считать универсальным.

### Заключение

Предложенный метод синтеза генераторов взвешенных кодов с суммированием позволяет получать простые структуры данных устройств посредством соединения полусумматоров и полных сумматоров. Путем дальнейшей замены типовых устройств на соединения мультиплексоров можно получать генераторы на основе коммутаторов каналов. Это открывает возможности в реализации генера-

торов кодов на современной программируемой элементной базе.

#### ЛИТЕРАТУРА

1. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Взвешенные коды с суммированием для организации контроля логических устройств // Электронное моделирование. – 2014. – Т. 36, №1. – С. 59-80.
2. Das D., Touba N.A. Weight-Based Codes and Their Application to Concurrent Error Detection of Multilevel Circuits // Proceedings of the 17th IEEE VLSI Test Symposium, USA, CA, Dana Point, 25-29 April. – 1999. – P. 370-376.
3. Das D., Touba N.A., Seuring M., Gossel M. Low Cost Concurrent Error Detection Based on Modulo Weight-Based Codes // Proceedings of IEEE 6th International On-Line Testing Workshop (IO LTW), Spain, Palma de Mallorca, July 3-5. – 2000. – P.171-176. DOI: 10.1109/OLT.2000.856633.
4. Ghosh S., Lai K.W., Jone W.B., Chang S.C. Scan Chain Fault Identification Using Weight-Based Codes for SoC Circuits // Proceedings of 13<sup>th</sup> Asian Test Symposium, 15-17 November. – 2004. – P. 210-215.
5. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В., Дмитриев В.В. Новые структуры систем функционального контроля логических схем // Автоматика и телемеханика. – 2017. – №2. – С. 127-143.
6. Bliudov A., Nazarov I., Dmitriev V., Kovalyov K. Use of Systematic Code Based on Data Bits Weighing for Concurrent Error Detection Considering Error Structure Analysis // Proceedings of 16th IEEE East-West Design & Test Symposium (EWDTS2018), Kazan, Russia, 14-17 September. – 2018. – P. 433-449.
7. Berger J.M. A Note on Error Detection Codes for Asymmetric Channels // Information and Control. – 1961. – Vol. 4, Issue 1. – P. 68-73. DOI: 10.1016/S0019-9958(61)80037-5.
8. Saposhnikov V., Saposhnikov V.I. New Code for Fault Detection in Logic Circuits // Proceedings of 4<sup>th</sup> International Conference on Unconventional Electromechanical and Electrical Systems, St. Petersburg, Russia, 21-24 June. – 1999. – P. 693-696.
9. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В., Дмитриев В.В. Свойства кодов с суммированием взвешенных переходов с прямой последовательностью весовых коэффициентов // Информатика и системы управления. – 2014. – №4. – С. 77-88.
10. Piestrak S.J. Design of Self-Testing Checkers for Unidirectional Error Detecting Codes. – Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 1995.
11. Дрозд А.В., Харченко В.С., Антощук 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 университет им. Н. Е. Жуковского «ХАИ», 2012.
12. Harris D.M., Harris S.L. Digital Design and Computer Architecture. – Morgan Kaufmann, 2012.
13. McCluskey E.J. Logic Design Principles: With Emphasis on Testable Semicustom Circuits. – N.J.: Prentice Hall PTR, 1986.
14. Lala P.K. Principles of Modern Digital Design. – New-Jersey: John Wiley & Sons, 2007.
15. Сапожников В.В., Сапожников Вл.В., Ефанов Д.В. Алгоритмы синтеза генераторов модульных кодов с суммированием взвешенных переходов с последовательностью весовых коэффициентов, образующих натуральный ряд чисел // Автоматика на транспорте. – 2017. – Т. 3, №2. – С. 280-301.
16. Ефанов Д.В. Способ синтеза генераторов взвешенных кодов с суммированием // Известия высших учебных заведений. Физика. – 2016. – Т. 59, №8/2. – С. 33-36.

17. *Kang M.* A Study of Self-Checking Circuit Design Based on Berger Code (in Chinese). – Master's dissertation, Harbin Engineering University, 2007.
18. *Sentovich E.M., Singh K.J., Moon C., Savoj H., Brayton R.K., Sangiovanni-Vincentelli A.* Sequential Circuit Design Using Synthesis and Optimization // Proceedings IEEE International Conference on Computer Design: VLSI in Computers & Processors, Cambridge, MA, USA, 11-14 October. – 1992. – P. 328-333. DOI: 10.1109/ICCD.1992.276282.
19. *Ефанов Д.В.* Генератор тестера кода с суммированием на новой элементной базе // Известия Петербургского университета путей сообщения. – 2010. – №1. – С. 86-92.
20. *Сапожников В.В., Сапожников Вл.В., Ефанов Д.В., Котенко А.Г.* Модульные коды с суммированием взвешенных переходов с последовательностью весовых коэффициентов, образующей натуральный ряд чисел // Труды СПИИРАН. – 2017. – №1. – С. 137-164. DOI: 10.15622/SP.50.6.
21. *Efanov D., Sapozhnikov V., Sapozhnikov V.I.* Generalized Algorithm of Building Summation Codes for the Tasks of Technical Diagnostics of Discrete Systems // Proceedings of 15<sup>th</sup> IEEE East-West Design & Test Symposium (EWDTS2017), Novi Sad, Serbia, September 29 – October 2, 2017, pp. 365-371, doi: 10.1109/EWDTS.2017.8110126.
22. *Сапожников В.В., Сапожников Вл.В., Ефанов Д.В.* Модульно-модифицированные взвешенные коды с суммированием, обнаруживающие любые ошибки нечетных кратностей // Электронное моделирование. – 2018. – Т. 40, №3. – С. 41-61. – DOI: 10.15407/emodel.40.03.041.
23. *Сапожников В.В., Сапожников Вл.В., Ургансков Д.И.* Универсальные структуры двоичных счетчиков единиц по произвольному модулю счета // Электронное моделирование. – 2002. – Т. 24, №4. – С. 65-81.
24. *Сапожников В.В., Сапожников Вл.В., Ургансков Д.И.* Блочная структура двоичного счетчика единиц по произвольному модулю счета // Электронное моделирование. – 2005. – Т. 27, №4. – С. 65-81.

*Статья представлена к публикации членом редколлегии С.В. Шалобановым.*

*E-mail:*

*Ефанов Дмитрий Викторович – TrES-4b@yandex.ru;*

*Тэн Тэн – liliya.ten@list.ru.*