Micro- and nanoelectronics. Condensed matter physics

Микро- и наноэлектроника. Физика конденсированного состояния

UDC 004.052.2
https://doi.org/10.32362/2500-316X-2023-11-5-54-62

RESEARCH ARTICLE

Application of double-error correction codes to protect configuration programmable logic memory against space radiation

Ekaterina S. Lepeshkina @**,**
Nikita D. Kustov,
Vladislav Kh. Khanov

Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, 660037 Russia
@ Corresponding author, e-mail: klepka1111.93@mail.ru

Abstract
Objectives. Programmable logic integrated circuits of the field programmable gate array (FPGA) type based on static configuration memory are widely used in the electronics of onboard spacecraft systems. Under the influence of space radiation, errors may occur in the FPGA configuration memory. The main methods of protection against such errors involve various options for reservation triggers, as well as the use of error-correcting codes in special error detection and correction circuits. The purpose of the present work is to determine which error-correcting codes are best suited to the implementation of internal scrubbing of the FPGA configuration memory taking redundancy into account.

Methods. The paper analyses various methods for scrubbing FPGA configuration memory, which are used to correct errors caused by the action of space radiation. It is proposed to increase the efficiency of internal scrubbing of the FPGA configuration memory using codes that correct both single- and double-adjacent SEC-DED-DAEC errors. In this case, the need to perform external scrubbing of the configuration memory is reduced by overwriting it with a reference configuration from non-volatile radiation-resistant memory; in this way, FPGA downtime caused by the external scrubbing procedure is reduced. Due to the known SEC-DED-DAEC codes having a non-zero probability of erroneous detection and subsequent erroneous correction of a double non-adjacent error, as well as various redundancy and implementation complexities, a study was made of the most efficient code for internal scrubbing.

Results. The results showed that the Datta, Neale and Hoyoon–Yongsurk codes are optimal from the indicated positions. Recommendations are given for selecting a specific code depending on the specific requirements for a particular planned space mission.

Conclusions. The study confirms the effectiveness of protecting the memory of programmable logic by using two-error-correcting codes.

Keywords: programmable logic integrated circuits, faults in configuration memory, methods for clearing configuration memory from faults, double-adjacent error-correcting codes

• **Submitted:** 30.01.2023 • **Revised:** 17.05.2023 • **Accepted:** 07.07.2023

For citation: Lepeshkina E.S., Kustov N.D., Khanov V.Kh. Application of double-error correction codes to protect configuration programmable logic memory against space radiation. *Russ. Technol. J.* 2023;11(5):54–62. https://doi.org/10.32362/2500-316X-2023-11-5-54-62

Financial disclosure: The authors have no a financial or property interest in any material or method mentioned.

The authors declare no conflicts of interest.

НАУЧНАЯ СТАТЬЯ

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

Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева, Красноярск, 660037 Россия
@ *Автор для переписки, e-mail: klepka1111.93@mail.ru*

Резюме

Цели. Программируемая логика типа field programmable gate array (FPGA) на основе статической конфигурационной памяти широко применяется в электронике бортовых систем космических аппаратов. Под воздействием космической радиации в конфигурационной памяти FPGA могут возникать ошибки. Основными методами защиты от них являются различные варианты резервирования триггеров, а также применение помехоустойчивых кодов в специальных схемах детектирования и исправления ошибок. Цель работы – определение из группы помехоустойчивых кодов тех, которые с учетом их избыточности наилучшим образом подходят для реализации внутреннего скраббинга конфигурационной памяти программируемых логических интегральных схем.

Методы. В работе рассмотрены методы скраббинга конфигурационной памяти FPGA, которые применяются для ее очистки от ошибок, вызванных действием космической радиации. Предлагается для повышения эффективности внутреннего скраббинга конфигурационной памяти FPGA использовать коды, исправляющие как однократные, так и двукратные смежные ошибки SEC-DED-DAEC. В этом случае уменьшается необходимость выполнения внешнего скраббинга конфигурационной памяти путем ее перезаписи эталонной конфигурацией из энергонезависимой радиационно-стойкой памяти. Таким образом, снижается время неработоспособного состояния FPGA, вызванное процедурой внешнего скраббинга. В связи с тем, что известные коды SEC-DED-DAEC имеют ненулевую вероятность ошибочного детектирования, а затем – ошибочного исправления двойной несмежной ошибки, а также обладают разной избыточностью и сложностью реализации, было приведено исследование наиболее эффективного кода для внутреннего скраббинга.

Результаты. Результаты исследования показали, что наилучшими с указанных позиций являются коды Датта, Нила и Хоюна – Йонгсурка. Приведены результаты сравнения кодов по выбранным критериям. Даны рекомендации для выбора конкретного кода в зависимости от возможных требований к планируемой космической миссии.

Выводы. Проведенное исследование показало эффективность защиты памяти программируемой логики с помощью применения кодов с исправлением двух ошибок.

Ключевые слова: программируемая логическая интегральная схема, сбой в конфигурационной памяти, методы очистки конфигурационной памяти от сбоев, коды с исправлением двух смежных ошибок

• **Поступила:** 30.01.2023 • **Доработана:** 17.05.2023 • **Принята к опубликованию:** 07.07.2023

Для цитирования: Лепёшкина Е.С., Кустов Н.Д., Ханов В.Х. Применение кодов с исправлением двух ошибок для защиты конфигурационной памяти программируемой логики от действия космической радиации. *Russ. Technol. J.* 2023;11(5):54–62. https://doi.org/10.32362/2500-316X-2023-11-5-54-62

Прозрачность финансовой деятельности: Авторы не имеют финансовой заинтересованности в представленных материалах или методах.

Авторы заявляют об отсутствии конфликта интересов.

INTRODUCTION

Currently, there is a significant progress in the development of space electronics, which has significantly expanded the functionality of modern spacecraft, while reducing the size, mass and power consumption of on-board systems. At the same time, the problem of protecting hardware from radiation failures in electronic components remains urgent, especially since cheap components from the commercial electronic component base (ECB), which lacks failure protection mechanisms typical of radiation-resistant space ECB, have begun to be widely used on small spacecraft. In hardware based on commercial ECBs, only single event upset (SEU) protection methods, which set out to correct single events of charged radiation particles hitting electronic components, are for the most part capable of application [1]. Such failures cause logic errors without destroying the component, i.e., they do not cause irreversible processes in the semiconductor structure of the component [2]. However, SEU failures tend to occur in response to triggers, with which any electronic equipment can be easily saturated. The main drawback of such an approach is the relatively high probability of such failures.

The main methods of protection against SEU failures are various options of redundancy, including triple modular redundancy (TMR) [3, 4] of the triggers, as well as the use of error correction codes (ECC) in special error detection and correction circuits (EDAC) [5, 6], which typically correct a single error (one-bit error) in the structure of several triggers, for example, in the SRAM word (static random access memory).

One of the widely used components in space instrumentation consists in the SRAM FPGA (static random access memory field programmable gate array) type of programmable logic integrated circuits. Such programmable logic integrated circuits (FPGAs), which mainly belong to the commercial category of microcircuits, are characterized by a large number of programmable elements, low power consumption, and a high speed of operation of the circuits implanted in them, but low fault tolerance to SEU events [7]. SEU errors can occur not only in the triggers of the SRAM-implemented FPGA circuit, but also in the SRAM configuration memory. The latter type of error is the most dangerous due to the change to the structure of the implemented circuit, resulting in constant periodic failures whenever signals pass through the damaged part of the circuit [8].

The main method of SRAM FPGA protection from SEU is scrubbing, which consists in overwriting the faulty contents of the configuration memory by the reference configuration whenever an error is detected in the implemented circuit [9, 10].

In this paper, various scrubbing variants are considered, and the possibilities of applying ECCs for scrubbing are analyzed. In this connection, the efficiency of the known SEC-DED-DAEC codes for the purpose of scrubbing SRAM FPGA configuration memory is evaluated assuming the application of SEC-DED-DAEC (single error correction, double error detection and double-adjacent error correction) type codes.

RELATED WORKS

Since the content of FPGA SRAM configuration memory can be corrupted by radiation exposure, configuration memory errors must be detected and corrected quickly in order to maintain correct operation in a radiation-exposed environment such as outer space. Configuration memory errors are eliminated by scrubbing, comprising a method for overwriting memory with a reference configuration.

There are two main overwriting approaches: full and partial. Full scrubbing consists in overwriting the entire contents of the configuration memory, while partial scrubbing comprises a block-by-block overwriting of the configuration memory [11]. In the second case, the system inoperability time due to loading the full configuration is reduced.

Scrubbing can also be “blind” and “sighted” [12]. “Blind” scrubbing consists in periodic overwriting of full or sequentially block configuration memory. Here, the overwriting period is selected from the anticipated error rate, for example, depending on the orbital altitude of the spacecraft. "Sight" scrubbing is performed following the detection of an error, possibly when the configuration memory block has been localized, errors have accumulated, and it is impossible to correct them by any other additional method.

There are direct and indirect approaches for detecting errors in the configuration memory. The indirect method consists in identifying errors in the internal FPGA system configured with the contents of the configuration memory. If there is a failure in the configuration memory, it will appear in the system in the form of an error, which can be used to detect (and rectify) TMR blocks distributed in the system. In case of a repeated (3–5 times consecutively) detected error, the scrubbing procedure should be started [11].

The direct method consists in periodic block-by-block comparison of the contents of the configuration memory with the corresponding block of the reference configuration, which is located in non-volatile and radiation-resistant memory. Here, ECCs can be used to accelerate the work. In this case, each block of the reference configuration in the non-volatile memory also stores its checksum. When the corresponding block is read back from the FPGA configuration memory, its checksum is calculated and compared with the corresponding sum from the non-volatile memory. If the sums do not match, it means that a failure has occurred in this block of the configuration memory and that this block must consequently be overwritten [8].

Modern Xilinx[1] FPGAs have built-in mechanisms based on the application of EDAC and single error correction and double error detection (SEC-DED) codes for block-by-block scanning of the configuration memory and automatic correction of single errors in it, as well as double errors detection in the scanned block [8]. This approach, further referred to in the paper as internal scrubbing, allows to reduce the number of runs of external partial scrubbing of configuration memory to correct double and larger multiplicity errors, and thus to reduce downtime of the FPGA-implemented system associated with the execution of external scrubbing.

PROBLEM STATEMENT

SEC-DED codes are known to fix one fault in a memory word. The SEC-DED code used must be low redundancy, fast, and easy to implement (for FPGA, it comprises a small number of logic elements for implementing a particular code). These criteria are best met by the Hsiao code (39, 32) [13] (where 39 is the codeword size in bits, 32 is informational), which belongs to the group of modified Hamming codes [14].

With the development of new ECB creation technologies and the transition to finer component production processes, the probability of multibit failures—primarily two-bit failures—increases [15]. One of the approaches to solving this problem consists in the use of SEC-DED-DAEC codes [16]. Codes belonging to this group correct 2 adjacent errors (adjacent means located in two adjacent bits of one memory word). Such codes also belong to the group of modified Hamming codes, meaning that they are fast and have low redundancy. Any SEC-DED-DAEC code copes well with the tasks of correcting single- and double adjacent errors. However, such codes have a specific disadvantage, consisting in the probability of incorrect correction of non-adjacent double errors; it can happen that one code from this group has a higher probability, while for another code, the probability is lower.

It should be noted that the probability of a contiguous double error in a memory word is significantly higher than the probability of a double non-contiguous error. A double contiguous error ensues from a single SEU event spanning two adjacent bits of the same word, while a double non-adjacent error comprises an accumulation of errors. First there is a failure that causes an error in one bit of a memory word, then, as time passes, another failure occurs causing an error in another bit of the same memory word, resulting in a double non-contiguous error. Clearly, although this process is unlikely, it cannot be ruled out. Therefore, it would be wrong to exclude from the SEC-DED-DAEC code analysis the estimation of the probability of incorrectly correcting unrelated double errors.

SEC-DED-DAEC codes are proposed as a means to improve the efficiency of internal scrubbing of the FPGA configuration memory. In this case, both single and double contiguous errors are corrected in the configuration memory to permit less recourse to external scrubbing of the configuration memory by overwriting it with a reference configuration from the non-volatile radiation-resistant memory.

The research task then appears as follows. Due to the fact that the number of SEC-DED-DAEC codes comprises some set, each of them has different parameters; therefore, it is necessary to conduct research in order to obtain an estimate of the probability of false detection (and subsequent false correction) of a twofold unrelated error along with the required resources for the implementation of the encoder/decoder, thus determining which of them are best fitted to implement internal FPGA configuration memory scrubbing given their additional redundancy.

RESEARCH PROVISION

As it was noted earlier, for SEC-DED-DAEC codes there is a probability of erroneous detection of double non-contiguous error as double contiguous, which

[1] https://www.xilinx.com/. Accessed February 16, 2023.

subsequently leads to erroneous correction of code word bits. The determination of such probability for this or that code is carried out by means of functional modeling.

For this purpose, a functional model was developed, presented as a program in C++, developed in the *Microsoft Visual Studio*[2] environment. The main logical parts of the program are described below:

1) variable initialization block (generating **G**-matrix and validating **H**-matrix);
2) information word generation block (one-dimensional Boolean array) using a pseudorandom function;
3) information word coding block by enumerating the columns of **G**-matrix. When one appears, an XOR operation is performed on the current bit of the codeword and the corresponding bit of the information word;
4) a block for introducing a double non-adjacent error using a pseudorandom function to determine random non-adjacent positions and subsequent inversion of the codeword bits;
5) error syndrome detection block by enumerating columns of the **H**-matrix. When one appears, an XOR operation is performed on the current bit of the syndrome and the corresponding bit of the codeword;
6) block of double adjacent error detection by comparing the resulting syndrome with the syndromes of double adjacent errors resulting from the XOR operation on two neighboring columns of the **H**-matrix;
7) block for detecting a double non-adjacent error. A double non-adjacent error is detected if a double adjacent error is not detected.

The algorithm goes through a large number of iterations. The probability of error-free detection of a double non-adjacent error is determined by the ratio of the number of outcomes of double non-adjacent error detection to the total number of code words with introduced double non-adjacent error that have passed through the algorithm.

In order to determine the resources required to implement the encoder and decoder codes (the number of logical elements), a simulation was performed. For this purpose, a program was developed using the VHDL language in the *Quartus*[3] development environment. Functional debugging was performed in the *ModelSim*[4] environment.

The logical parts of the encoder are organized as follows:

1) initialization block for variables (CLK clock signal, input data word and output codeword);
2) coding block. When the CLK signal changes, a cycle of encoding the checksum bits according to the **G**-matrix is started using the XOR operation;
3) codeword output block (initial information word and control sum).

Logical parts of a decoder:

1) initialization block for variables (CLK clock signal, input codeword, output corrected word);
2) error syndrome detection block. When the CLK signal changes, the loop for the codeword bits is started; the error syndrome is calculated according to the **H**-matrix;
3) decoding block. Firstly, the case without an error is checked (if the syndrome is zero). If the syndrome is non-zero, the resulting syndrome is compared to single error syndromes and error correction is performed. If there are no coincidences with single error syndromes, comparison with double adjacent error syndromes and error correction takes place. If there are no matches with syndromes, a non-adjacent error is detected;
4) block of corrected word output.

Simulation was performed for FPGA Cyclone IV E EP4CE6E22A7 (Intel, USA). As a result of the synthesis, the number of used encoder and decoder logic elements was counted.

Functional and simulation models for several codes were developed in accordance with the presented description. The main criterion for the selection of the codes under study is the explicit description in the scientific and technical literature of the H-matrices of the codes. In this case, the probability of incorrect generation of the check matrix is excluded, the research process is simplified, and the probability of an error occurring in the simulated results is minimized. Thus, the following codes with 32-bit information word were chosen as the codes under study: Dutta (39, 32) [17], Datta (42, 32) [18], Neale (42, 32) [19], Reviriego (39, 32) [20], Cha–Yoon (39, 32) [21], Hoyoon–Yongsurk (41, 32) [22].

In addition to the SEC-DED-DAEC codes, we present for comparison data for one SEC-DED code, namely, the Hsiao code [13], which is widely used in the implementation of the EDAC mechanism for memory.

RESULTS

The results of the studies are presented in the table. The given data show that the leading positions are occupied by the Datta, Neale and Hoyoon–Yongsurk codes. While the Hsiao code shows the average values

[2] https://visualstudio.microsoft.com/ru/ (in Russ.). Accessed February 16, 2023.

[3] http://altera.ru/soft_quartus.html. Accessed February 16, 2023.

[4] https://altera.co.uk/products/software/quartus-ii/modelsim/qts-modelsim-index.html. Accessed February 16, 2023.

for the selected comparison criteria, it should be kept in mind that it does not correct two adjacent errors. Comparing them with each other in relation to the use of FPGA configuration SRAM memory scanning, the following recommendations can be made.

Table. Comparison of SEC-DED-DAEC codes according to the criteria of error-free detection of a double non-adjacent error and implementation complexity

Code	Probability of error-free detection of a double non-adjacent error	Number of encoder/decoder logical elements
Hsiao (39, 32)	63.4	164/332
Dutta (39, 32)	43.5	170/407
Datta (40, 32)	78.9	164/354
Neale (42, 32)	84.4	57/391
Reviriego (39, 32)	38.4	175/373
Cha–Yoon (39, 32)	60.7	123/272
Hoyoon–Yongsurk (41, 32)	95.7	178/384

For the Datta code, the redundancy of the codeword is aligned to the byte dimension, which can simplify the construction of the FPGA configuration memory. While the probability of error detection of double non-adjacent error is the worst of the considered three codes, it is slightly inferior to the Neale code. The complexities of the coder and decoder code are low; the complexity of the coder is average.

For the Neale code, the redundancy of the codeword is not aligned with the byte boundary. The probability of erroneous detection of double non-adjacent error is average, but at a rather high level. Encoder complexity is very low; decoder complexity is the highest.

The redundancy for the Hoyoon–Yongsurk code is also not aligned with the byte boundary. Here, while the probability of error detection of double non-adjacent error is the best and encoder complexity is the highest, decoder complexity is average.

The selection of one or another code choice for implementation should take into account the initial data for the planned space mission. If simplicity and speed of implementation are critical with an acceptable probability of false detection of double non-adjacent error under conditions of low intensity of radiation failures, the Datta configuration memory code can be recommended for use in the EDAC mechanism. The Neale code can also qualify for this position, but it has the highest redundancy, and is not aligned on the byte boundary. If the planned space mission is long-term and will be carried out under conditions of a high failure rate, it may be better to use the Hoyoon–Yongsurk code, despite its relative implementation complexity and low performance.

In addition, we note that the complexity of the analyzed triplet codes, expressed in the number of required logical elements for the implementation of the encoder/decoder, on average corresponds to the complexity of the commonly used Hsiao code. Therefore, problems do not arise in terms of the use of FPGA resources with their implementation for the purpose of configuration memory scanning when scrubbing.

CONCLUSIONS

The present work proposes that the algorithm of the internal scanning method be changed for the internal scrubbing of the FPGA SRAM configuration memory: instead of one of the SEC-DED codes, such as the Hsiao code, it is proposed to use one of the SEC-DED-DAEC codes. In this case, both a single error and two adjacent errors will be corrected in the configuration SRAM, which will reduce the number of external partial configuration memory scrubbing runs to correct double and larger multiplicity errors, thus reducing the downtime of the FPGA-implemented system associated with scrubbing implementation.

SEC-DED-DAEC codes have one negative property: non-zero probability of erroneous detection (and then erroneous correction) of double non-adjacent error. In addition, they have different redundancies for storing the checksum in SRAM memory, as well as various complexities of implementation, which can be estimated by the required number of logical elements to implement the encoder/decoder. A study was carried out to determine the most efficient code according to these criteria among SEC-DED-DAEC codes with known check matrices. The results of the study showed that the Datta, Neale, and Hoyoon–Yongsurk codes are optimal from these positions. While the Hoyun–Yongsurk code has almost zero probability of error detection of double adjacent error, at the same time, it has the greatest complexity. The Dutta code is the easiest to implement, but the probability of detecting a double non-adjacent error in error is about 20%. The Neale code occupies an intermediate position. When selecting error correction approaches, the choice of a particular code should be determined by the requirements of the planned space mission.

Acknowledgments

The reported study was supported by the Russian Foundation for Basic Research, project No. 19-38-90052.

Authors' contribution. All authors equally contributed to the research work.

REFERENCES

1. Maksimenko S.L., Melekhin V.F., Filippov A.S. Analysis of the problem of radiation-tolerant information and control-systems implementation. *Informatsionno-upravlyayushchie sistemy = Information and Control Systems.* 2012;2(57):18–25 (in Russ.). Available from URL: http://www.i-us.ru/index.php/ius/article/view/13788
2. Gaillard R. Single Event Effects: Mechanisms and Classification. In: Nicolaidis M. (Ed.). *Soft Errors in Modern Electronic Systems. Frontiers in Electronic Testing.* Boston, MA: Springer; 2011. V. 41. P. 27–54. https://doi.org/10.1007/978-1-4419-6993-4_2
3. Kastensmidt F.L., Sterpone L., Carro L., Reorda M.S. On the Optimal Design of Triple Modular Redundancy Logic for SRAM-based FPGAs. In: *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*. 2005. P. 1290–1295. https://doi.org/10.1109/DATE.2005.229
4. Cheng D., Qi D., Chen M. Radiation-hardened Test Design for Aerospace SoC. In: *2020 IEEE 5th International Conference on Integrated Circuits and Microsystems* (*ICICM*). 2020. P. 213–217. https://doi.org/10.1109/ICICM50929.2020.9292308
5. Mang I., Mang E., Popescu C. VHDL implementation of an error detection and correction module based on Hamming code. *J. Comput. Sci. Control Syst.* 2011;4(2):43–46.
6. Baviera E., Schettino G.M., Tuniz E., Vatta F. Software Implementation of Error Detection and Correction Against Single-Event Upsets. In: *2020 International Conference on Software, Telecommunications and Computer Networks* (*SoftCOM*). 2020. https://doi.org/10.23919/SoftCOM50211.2020.9238173
7. Maksfild K. *Proektirovanie na PLIS. Arkhitektura, sredstva i metody: kurs molodogo boitsa* (*The Design Warrior's Guide to FPGAs: Devices, Tools and Flows*.): transl. from Engl. Moscow: Dodeka-XXI; 2007. 408 p. (in Russ.).
[Maxfield C. *The Design Warrior's Guide to FPGAs: Devices, Tools and Flows*. Oxford, UK: Jordan Hill; 2004. 560 p.]
8. Virtlin M., Khardint A. Hybrid Clear Configuration for Xilinx FPGAs. In: *PLIS i parallel'nye struktury v aerokosmicheskoi oblasti. Programmnye oshibki i otkazoustoichivoe proektirovanie* (*FPGAs and Parallel Architectures for Aerospace Applications. Soft Errors and Fault-Tolerant Design*). Kastensmidt F., Rech P. (Eds.). Moscow: Tekhnosfera; 2018. 326 p. (in Russ.).
9. Sokolov A. Software and hardware methods of radiation hardening of microprocessors based control systems. *Sovremennaya Elektronika*. 2014;6:30–33 (in Russ.).
10. Zhang R., Xiao L., Li J., Cao X., Li L. An Adjustable and Fast Error Repair Scrubbing Method Based on Xilinx Essential Bits Technology for SRAM-Based FPGA. *IEEE Transactions on Reliability*. 2020;69(2):430–439. https://doi.org/10.1109/TR.2019.2896897
11. Tsetin E., Dizzel' O., Li T., et al. Review and study of methods for detecting and eliminating single failures for heterogeneous systems based on FPGAs. In: *PLIS i parallel'nye struktury v aerokosmicheskoi oblasti. Programmnye oshibki i otkazoustoichivoe proektirovanie* (*FPGAs and Parallel Architectures for Aerospace Applications. Soft Errors and Fault-Tolerant Design*).

СПИСОК ЛИТЕРАТУРЫ

1. Максименко С.Л., Мелехин В.Ф., Филиппов А.С. Анализ проблемы построения радиационно-стойких информационно-управляющих систем. *Информационно-управляющие системы*. 2012;2(57):18–25. URL: http://www.i-us.ru/index.php/ius/article/view/13788
2. Gaillard R. Single Event Effects: Mechanisms and Classification. In: Nicolaidis M. (Ed.). *Soft Errors in Modern Electronic Systems. Frontiers in Electronic Testing.* Boston, MA: Springer; 2011. V. 41. P. 27–54. https://doi.org/10.1007/978-1-4419-6993-4_2
3. Kastensmidt F.L., Sterpone L., Carro L., Reorda M.S. On the Optimal Design of Triple Modular Redundancy Logic for SRAM-based FPGAs. In: *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*. 2005. P. 1290–1295. https://doi.org/10.1109/DATE.2005.229
4. Cheng D., Qi D., Chen M. Radiation-hardened Test Design for Aerospace SoC. In: *2020 IEEE 5th International Conference on Integrated Circuits and Microsystems* (*ICICM*). 2020. P. 213–217. https://doi.org/10.1109/ICICM50929.2020.9292308
5. Mang I., Mang E., Popescu C. VHDL implementation of an error detection and correction module based on Hamming code. *J. Comput. Sci. Control Syst.* 2011;4(2):43–46.
6. Baviera E., Schettino G.M., Tuniz E., Vatta F. Software Implementation of Error Detection and Correction Against Single-Event Upsets. In: *2020 International Conference on Software, Telecommunications and Computer Networks* (*SoftCOM*). 2020. https://doi.org/10.23919/SoftCOM50211.2020.9238173
7. Максфилд К. *Проектирование на ПЛИС. Архитектура, средства и методы: курс молодого бойца*: пер. с англ. М.: Додэка-XXI; 2007. 408 с.
8. Виртлин М., Хардинт А. Гибридная очистка конфигурации для ПЛИС компании Xilinx. В кн.: *ПЛИС и параллельные структуры в аэрокосмической области. Программные ошибки и отказоустойчивое проектирование*; под ред. Ф. Канстеншмидт, П. Реха. М.: Техносфера; 2018. 326 с.
9. Соколов А. Программно-аппаратные методы повышения радиационной стойкости микросхем SRAM ПЛИС. *Современная электроника*. 2014;6:30–33.
10. Zhang R., Xiao L., Li J., Cao X., Li L. An Adjustable and Fast Error Repair Scrubbing Method Based on Xilinx Essential Bits Technology for SRAM-Based FPGA. *IEEE Transactions on Reliability*. 2020;69(2):430–439. https://doi.org/10.1109/TR.2019.2896897
11. Цетин Э., Диззель О., Ли Т. и др. Обзор и исследование методов обнаружения и устранения одиночных сбоев для гетерогенных систем на основе ПЛИС. В кн.: *ПЛИС и параллельные структуры в аэрокосмической области. Программные ошибки и отказоустойчивое проектирование*; под ред. Ф. Канстеншмидт, П. Реха. М.: Техносфера; 2018. 326 с.
12. Флеминг Ш.Т., Томас Д.В., Винтерстейн Ф. Энергосберегающая адаптивная платформа FDIR, применяющая модули гетерогенных систем на кристалле. В кн.: *ПЛИС и параллельные структуры в аэрокосмической*

Kastensmidt F., Rech P. (Eds.). Moscow: Tekhnosfera; 2018. 326 p. (in Russ.).

12. Fleming Sh., Tomas D., Vintersteyn F. Energy-saving adaptive FDIR platform using heterogeneous system-on-chip modules. In book: *PLIS i parallel'nye struktury v aerokosmicheskoi oblasti. Programmnye oshibki i otkazoustoichivoe proektirovanie* (*FPGAs and Parallel Architectures for Aerospace Applications. Soft Errors and Fault-Tolerant Design*). Kastensmidt F., Rech P. (Eds.). Moscow: Tekhnosfera; 2018. 326 p. (in Russ.).
13. Hsiao M.Y. A class of optimal minimum odd-weight-column SEC-DED codes. *IBM J. Res. Develop*. 1970;14(4):395–401. https://doi.org/10.1147/rd.144.0395
14. Hamming R.W. Error detecting and error correcting codes. *Bell System technical journal*. 1950;29(2):147–160. Available from URL: https://www.sci-hub.ru/10.1002/j.1538-7305.1950.tb00463.x
15. Chumakov A.I., Sogoyan A.V., Boruzdina A.B., et al. Mechanisms of multiple cell upsets in memory. *Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem* (*MES*) *= Problems of Advanced Micro- and Nanoelectronic Systems Development.* 2016;4:145–152 (in Russ.).
16. Krasnyuk A.A., Petrov K.A. Application features of the error correction coding in sub-100-nm memory microcircuits for cosmic systems. *Russ. Microelectron.* 2013;42(1):53–58. https://doi.org/10.1134/S1063739712040087
[Original Russian Text: Krasnyuk A.A., Petrov K.A. Application features of the error correction coding in sub-100-nm memory microcircuits for cosmic systems. *Mikroelektronika.* 2012;41(6):450–456 (in Russ.).]
17. Dutta A., Touba N.A. Multiple Bit Upset Tolerant Memory Using a Selective Cycle Avoidance Based SEC-DED-DAEC Code. In: *25th IEEE VLSI Test Symposium* (*VTS'07*). 2007. P. 349–354. https://doi.org/10.1109/VTS.2007.40
18. Datta R., Touba N.A. Exploiting unused spare columns to improve memory ECC. In: *27th IEEE VLSI Test Symposium*. 2009. P. 47–52. https://doi.org/10.1109/VTS.2009.52
19. Neale A., Sachdev M. A new SEC-DED error correction code subclass for adjacent MBU tolerance in embedded memory. *IEEE Transactions on Device and Materials Reliability.* 2013;13(1):223–230. https://doi.org/10.1109/TDMR.2012.2232671
20. Reviriego P., Liu S.S., Sánchez-Macián A., Xiao L., Maestro J.A. Unequal error protection codes derived from SEC-DED codes. *Electron. Lett*. 2016;52(8):619–620. https://doi.org/10.1049/el.2016.0077
21. Cha S., Yoon H. Efficient implementation of single error correction and double error detection code with check bit pre-computation for memories. *JSTS: J. Semiconductor Technol. Sci.* 2018;12(4):418–425. https://doi.org/10.5573/JSTS.2012.12.4.418
22. Hoyoon Jun, Yongsurk Lee. Protection of On-chip Memory Systems against Multiple Cell Upsets Using Double-adjacent Error Correction Codes. *Int. J. Computer Inform. Technol*. 2014;3(6):1316–1320. Available from URL: https://www.ijcit.com/archives/volume3/issue6/Paper030621.pdf

области. Программные ошибки и отказоустойчивое проектирование; под ред. Ф. Канстеншмидт, П. Реха. М.: Техносфера; 2018. 326 с.

13. Hsiao M.Y. A class of optimal minimum odd-weight-column SEC-DED codes. *IBM J. Res. Develop*. 1970;14(4):395–401. https://doi.org/10.1147/rd.144.0395
14. Hamming R.W. Error detecting and error correcting codes. *Bell System technical journal*. 1950;29(2):147–160. URL: https://www.sci-hub.ru/10.1002/j.1538-7305.1950.tb00463.x
15. Чумаков А.И., Согоян А.В., Боруздина А.Б. и др. Механизмы многократных сбоев в микросхемах памяти. *Проблемы разработки перспективных микро- и наноэлектронных систем* (*МЭС*). 2016;4:145–152. URL: http://www.mes-conference.ru/data/year2016/pdf/D188.pdf
16. Краснюк А.А., Петров К.А. Особенности применения методов помехоустойчивого кодирования в суб-100-нм микросхемах памяти для космических систем. *Микроэлектроника*. 2012;41(6):450–456.
17. Dutta A., Touba N.A. Multiple Bit Upset Tolerant Memory Using a Selective Cycle Avoidance Based SEC-DED-DAEC Code. In: *25th IEEE VLSI Test Symposium* (*VTS'07*). 2007. P. 349–354. https://doi.org/10.1109/VTS.2007.40
18. Datta R., Touba N.A. Exploiting unused spare columns to improve memory ECC. In: *27th IEEE VLSI Test Symposium*. 2009. P. 47–52. https://doi.org/10.1109/VTS.2009.52
19. Neale A., Sachdev M. A new SEC-DED error correction code subclass for adjacent MBU tolerance in embedded memory. *IEEE Transactions on Device and Materials Reliability.* 2013;13(1):223–230. https://doi.org/10.1109/TDMR.2012.2232671
20. Reviriego P., Liu S.S., Sánchez-Macián A., Xiao L., Maestro J.A. Unequal error protection codes derived from SEC-DED codes. *Electron. Lett*. 2016;52(8):619–620. https://doi.org/10.1049/el.2016.0077
21. Cha S., Yoon H. Efficient implementation of single error correction and double error detection code with check bit pre-computation for memories. *JSTS: J. Semiconductor Technol. Sci.* 2018;12(4):418–425. https://doi.org/10.5573/JSTS.2012.12.4.418
22. Hoyoon Jun, Yongsurk Lee. Protection of On-chip Memory Systems against Multiple Cell Upsets Using Double-adjacent Error Correction Codes. *Int. J. Computer Inform. Technol*. 2014;3(6):1316–1320. URL: https://www.ijcit.com/archives/volume3/issue6/Paper030621.pdf

About the authors

Ekaterina S. Lepeshkina, Assistant, Information Technology Security Department; Engineer, “Small Satellite” Laboratory, Reshetnev Siberian State University of Science and Technology (Reshetnev University). (31, Krasnoyarsky Rabochy pr., Krasnoyarsk, 660037 Russia). E-mail: klepka1111.93@mail.ru. Scopus Author ID 57218577296, RSCI SPIN-code 8746-6555, https://orcid.org/0000-0001-5116-6260

Nikita D. Kustov, Assistant, Information Technology Security Department; Engineer, “Small Satellite” Laboratory, Reshetnev Siberian State University of Science and Technology (Reshetnev University). (31, Krasnoyarsky Rabochy pr., Krasnoyarsk, 660037 Russia). E-mail: kustovnd@yandex.ru. Scopus Author ID 57218577358, RSCI SPIN-code 9031-4516, https://orcid.org/0000-0002-3362-3971

Vladislav Kh. Khanov, Cand. Sci. (Eng.), Associate Professor, Information Technology Security Department; Head of the “Small Satellite” Laboratory, Reshetnev Siberian State University of Science and Technology (Reshetnev University). (31, Krasnoyarsky Rabochy pr., Krasnoyarsk, 660037 Russia). E-mail: khvkh@mail.ru. Scopus Author ID 56491191500, RSCI SPIN-code 5197-5699, https://orcid.org/0000-0001-6720-9405

Об авторах

Лепёшкина Екатерина Сергеевна, ассистент, кафедра безопасности информационных технологий; инженер, лаборатория «Малые космические аппараты», Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева (660037, Россия, Красноярск, просп. им. газеты «Красноярский рабочий», д. 31). E-mail: klepka1111.93@mail.ru. Scopus Author ID 57218577296, SPIN-код РИНЦ 8746-6555, https://orcid.org/0000-0001-5116-6260

Кустов Никита Дмитриевич, ассистент, кафедра безопасности информационных технологий; инженер, лаборатория «Малые космические аппараты», Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева (660037, Россия, Красноярск, просп. им. газеты «Красноярский рабочий», д. 31). E-mail: kustovnd@yandex.ru. Scopus Author ID 57218577358, SPIN-код РИНЦ 9031-4516, https://orcid.org/0000-0002-3362-3971

Ханов Владислав Ханифович, к.т.н., доцент, кафедра безопасности информационных технологий; заведующий лабораторией «Малые космические аппараты», Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева (660037, Россия, Красноярск, просп. им. газеты «Красноярский рабочий», д. 31). E-mail: khvkh@mail.ru. Scopus Author ID 56491191500, SPIN-код РИНЦ 5197-5699, https://orcid.org/0000-0001-6720-9405

Translated from Russian into English by Lyudmila O. Bychkova
Edited for English language and spelling by Thomas A. Beavitt