ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.056.53 DOI https://doi.org/10.35546/kntu2078-4481.2025.4.3.1

С. Г. АНТОЩУК
доктор технічних наук, професор, директор
Інститут комп'ютерних систем
Національного університету «Одеська політехніка»
ORCID: 0000-0002-9346-145X

О. М. ІВАНОВА
старший викладач кафедри комп'ютерних систем
Національний університет «Одеська політехніка»
ORCID: 0000-0002-4743-6931

ШВИДКА ЛОКАЛІЗАЦІЯ ОСЕРЕДКІВ ПРОГРАМНОГО КОДУ FPGA, ПРИДАТНИХ ДЛЯ СТЕГАНОГРАФІЧНОГО ЗБЕРІГАННЯ ДОДАТКОВИХ ДАНИХ

Статтю присвячено використанню стеганографічного підходу до зберігання контрольних даних, які використовуються в процесі прихованого моніторингу характеристик безпеки програмного коду мікросхем FPGA. За такого підходу контрольні дані утворюють з інформаційним об'єктом програмного коду єдине ціле, а сам факт присутності цих даних в програмному коді не є очевидним. Традиційно для стеганографічного вбудовування додаткових даних в програмний код FPGA застосовують методи, що базуються на еквівалентних перетвореннях елементів програмного коду. Однак обсяг зберігання даних, забезпечений зазначеними методами часто є недостатнім для одночасного зберігання контрольних даних декількох видів моніторингу. Інша група методів стеганографічного зберігання даних базується на нееквівалентних перетвореннях програмного коду. Ці методи застосовуються для обчислювачів, реалізованих в середовищі FPGA, які виконують наближену обробку даних. В структурі зазначених обчислювачів мають місце елементарні блоки FPGA, які обчислюють тільки несуттєві розряди результатів наближених операцій. Зміна в нееквівалентний спосіб програмного коду таких блоків не призводить до некоректного функціонування пристрою. Відомі методи даного класу основані на моделюванні, яке знаходять всі несуттєві блоки, але має значну часову складність. В даній роботі пропонується підхід, який дозволяє знайти неповну множину несуттєвих блоків, але зі значним скороченням часової складності. Експериментальне дослідження запропонованого підходу показує, що його використання дозволяє забезпечити додатковий до методів еквівалентних перетворень обсяг стеганографічного зберігання контрольних даних в середовищі програмного коду FPGA. Пропозиції роботи можуть бути застосовані до обчислювачів, розмірність яких не дозволяє виконати локалізацію несуттєвих існуючими методами.

Ключові слова: *стеганографічне зберігання даних, FPGA, наближена обробка даних, операції над числами з плаваючою крапкою, блоки LUT.*

S. G. ANTOSCHUK
Doctor of Technical Sciences, Professor, Director
Institute of Computer Systems
of Odesa Polytechnic National University
ORCID: 0000-0002-9346-145X

O. M. IVANOVA
Senior Lecturer at the Department of Computer Systems
Odesa Polytechnic National University
ORCID: 0000-0002-4743-6931

FAST LOCALIZATION OF FPGA PROGRAM CODE ELEMENTS SUITABLE FOR STEGANOGRAPHIC STORAGE OF ADDITIONAL DATA

This article is devoted to the use of a steganographic approach to storing control data used in the process of hidden monitoring of the security characteristics of FPGA program code. With this approach, the control data forms a single whole with the information object of the program code, and the very fact of the presence of this data in the program code is not obvious. Traditionally, methods based on equivalent transformations of software code elements are used for steganographic embedding of additional data into FPGA program code. However, the volume of data storage provided by such methods is often insufficient for the simultaneous storage of control data for several types of monitoring. Another group of steganographic data storage methods is based on non-equivalent transformations of program code. These methods are used for computing elements implemented in an FPGA environment that perform approximate data processing. The structure of these computing elements includes elementary FPGA units that calculate only insignificant digits of the results of approximate operations. The program code of such units can be changed in a non-equivalent way without causing the device to malfunction. Known methods of this class are based on simulation, which finds all insignificant units, but they are significantly time complexity. This paper proposes an approach that allows finding an incomplete set of insignificant blocks, but with a significant reduction in time complexity. An experimental study of the proposed approach shows that its use allows for an increase in the volume of steganographic storage of control data in the FPGA software code environment. The proposals of this work are applicable to computing elements whose complexity does not allow for the localization of insignificant elementary units using existing methods.

Key words: *steganographic data storage, FPGA, approximate data processing, operations on floating-point numbers, LUT units.*

Постановка проблеми

Мікросхеми FPGA є програмно-керованими цифровими пристроями. Подібно мікропроцесорам та мікроконтролерам їх поведінка може змінюватися під впливом програмного коду. На відміну від мікропроцесорів, програмний код змінює структуру зав'язків та функції елементарних блоків FPGA тим самим налаштовуючи поведінку цих пристроїв. Паралельна структура FPGA створює можливість для отримання значно більшої продуктивності обчислень ніж продуктивність, доступна для мікропроцесорів.

FPGA є типовою елементною базою для високопродуктивних обчислень в галузях критичного застосування: енергетика, швидкісний транспорт, комунікації [1]. Критичність цих галузей підвищує ризики, пов'язані з зловмисним втручанням в функціонування комп'ютерних систем на базі FPGA. Втручання на апаратному рівні може бути реалізоване при наявності фізичного доступу до комп'ютерної систем. В той же час для втручання на програмному рівні достатньо обмеженого фізичного або віддаленого доступу. Зазначені фактори обумовлюють важливість процесів забезпечення цілісності та автентичності програмного коду FPGA для систем підвищеного ризику.

Одним з ефективних підходів до запобігання втручання в функціонування систем на базі FPGA є оперативний моніторинг цілісності та автентичності програмного коду цих мікросхем [2]. Однак проблемним моментом процедур моніторингу є місце та спосіб зберігання контрольних даних, які використовуються в процесі моніторингу. В даній роботі розглядається підхід до зберігання контрольних даних, який полягає в прихованому їх вбудовуванні в програмний код FPGA в стеганографічний спосіб.

Аналіз останніх досліджень і публікацій

Одним із ефективних підходів до зберігання контрольних даних є стеганографічний підхід. Найбільшого свого розвитку цей підхід отримав в застосуванні до мультимедійних стеганографічних інформаційних контейнерів, таких як растрові зображення, цифрове відео та цифровий звук [3]. Це пояснюється тим, що зазначені контейнери мають аналогову природу походження, а їх елементарних одиниць (пікселі, семпли) мають наближене подання. В результаті незначні спотворення цих даних не тільки не призводять до деградації контейнера, але й зазвичай не реєструється.

На відміну від мультимедійних інформаційних контейнерів, програмний код (і зокрема програмний код мікросхем FPGA) є інформаційним контейнером з точно поданими даними. Спотворення бітів програмного коду призводить до спотворення функціонування пристрою, що програмується цим кодом. Тому для стеганографічного вбудовування додаткових даних у програмний код FPGA зазвичай використовують методи еквівалентних перетворень [4]. Ці методи не змінюють функціонування мікросхеми FPGA та обсяг програмного коду, проте дозволяють приховано вбудувати в цей код додаткові дані.

Методи стеганографічного вбудовування даних у програмний код FPGA, які використовують еквівалентні перетворення є практичними, проте забезпечують малий, потенційно доступний для зберігання обсяг додаткових даних. Це часто призводить до дефіциту обсягу для зберігання контрольних даних у задачах оперативного моніторингу програмного коду FPGA, а також зменшує кількість різновидів моніторингу, що можуть бути застосовані до програмного коду.

Але для точно поданого інформаційного контейнера, яким є програмний код FPGA, також можливе стеганографічне вбудовування додаткових даних, подібне до того, що використовується в інформаційних контейнерах з наближено поданими елементарними одиницями. Зазначена можливість полягає в тому, що при реалізації наближеної обробки даних на FPGA, ця наближеність опосередковано переноситься на точно поданий програмний код. При виконанні наближених обчислень частина елементарних блоків FPGA беруть участь у обчисленні лише несуттєвих розрядів результатів обчислень [5]. Програмний код таких елементарних блоків може бути змінений в нееквівалентний спосіб для вбудовування додаткових даних. Причому таке нееквівалентне вбудовування не відбивається на коректності виконання функції пристрою.

Основними елементарними блоками структури мікросхем FPGA є блоки LUT (Look Up Table). У сучасних серіях FPGA кількість таких блоків становить мільйони та десятки мільйонів. Блок LUT має m входів та виконує обчислення однієї логічної функції від m змінних. Блоки LUT є програмованими блоками та налаштуються на реалізацію конкретної логічної функції 2^m бітним двійковим програмним кодом. При реалізації зазначених вище наближених арифметичних операцій на FPGA в її структурі можливо виділити елементарні обчислювальні блоки LUT, які беруть участь у формуванні тільки несуттєвих розрядів результату. В подальшому такі блоки LUT будуть називатися несуттєвими блоками. В силу цього програмний код таких блоків може бути модифікований і така модифікація не буде мати впливу на результат роботи пристрою та його параметри.

В роботах [6, 7] запропоновано підхід до пошуку несуттєвих блоків LUT в структурі FPGA-проєктів, які мають в своєму складі модулі наближеної обробки даних. Цей підхід дозволяє достатньо швидко знаходити несуттєві блоки для обчислювачів, в яких сукупна розрядність операндів не перевищує 34-36 розрядів. Для пошуку несуттєвих блоків в обчислювачах з більшою розрядністю пошук може потребувати значних часових ресурсів. Враховуючи цей недолік існуючого підходу, в даній статті пропонується не базований на переборі варіантів спосіб пошуку несуттєвих блоків, який має меншу часову складність та здатність виявити більшу частину несуттєвих блоків.

Формулювання мети статті

Розробити підхід до не базованого на переборі варіантів пошуку несуттєвих блоків LUT в структурі FPGA-проєктів, що реалізують наближену обробку даних. На основі цього підходу оцінити обсяг даних, які можливо вбудувати в програмний код FPGA в стеганографічний спосіб, використовуючи несуттєві блоки LUT.

Викладення основного матеріалу дослідження

В роботах [6, 7] запропоновано для локалізації несуттєвих блоків LUT виконувати пошук суттєвих блоків, інвертування значень на виході яких дає помилки в суттєвих розрядах результату обчислень. Однак цей підхід дозволяє швидко локалізувати потрібні блоки тільки в складі обчислювачів сукупна розрядність операндів яких не перевищує 34-36 розрядів. В даній роботі пропонується підхід, що не має зазначеного обмеження. Підхід базується на врахуванні структурних та арифметичних особливостей побудови обчислювачів, в складі яких виконується пошук несуттєвих блоків.

За умови вимоги про відповідність формату результату форматам операндів, обчислювач при наближеній обробці даних в частині обробки мантис за n-розрядними операндами отримує повний $2n$-розрядний результат з послідуючим округленням та відкиданням молодших n-розрядів. Відомим є метод скорочення структури матриці помноження мантис для описаного випадку [8]. Відповідно до цього методу матриця помноження розбивається на старшу та молодшу частини. Молодша частина матриці складається з k-стовбців. Метод скорочення структури матриці помноження обумовлює виключення молодшої частини матриці зі схеми. При цьому старша частина матриці обчислює усічений $2n–k$ -розрядний результат $V(2n…k+1)$, старші n-розрядів якого складають кінцевий результат $V(2n…n+1)$, формат якого відповідає формату операндів.

В зазначеному методі [8] величина k визначається за умови відсутності впливу молодшої частини схеми матриці на округлений кінцевий результат $V(2n…n+1)$. Значення k розраховуються виходячи з максимального значення складової повного результату, яку створює молодша частина матриці. Таке максимальне значення утворюється коли вся молодша частина матриці кон'юнкцій добутку заповнена одиницями. Відповідно до положень методу [8] значення k приближено оцінюється за формулою:

$$k = n - log_2 n;$$

де n – розмірність операндів та кінцевого результату.

Зазначені положення методу скорочення матриці помноження мантис пропонується використати для встановлення нижньої оцінки кількості несуттєвих блоків LUT та їх локалізації в структурі обчислювача. Блоки LUT, які відповідають молодшим k-стовбцям матриці в структурі обчислювача є несуттєвими через відсутність впливу значень на виходах цих блоків на округлений кінцевий результат. Будь-які спотворення значень на виходах зазначених блоків не приводять до змін в суттєвих розрядах результату та розрядах кінцевого результату обчислення. Аналіз структури матриці обчислювача у вигляді списків блоків LUT, а також списків зв'язків цих блоків між собою і з входами та виходами обчислювача дає можливість виявити сукупність блоків, які відносяться до молодшої частини матриці обчислювача.

Для визначення множини суттєвих блоків LUT та оцінки кількості несуттєвих блоків в роботах [6, 7] було запропоновано та розроблено два програмних додатки:

1. Додаток LUTListExtractor, призначення якого полягає в добуванні детальної інформації про структуру FPGA-проєкту з внутрішньої бази даних САПР Intel/Altera Quartus Prime. Ця детальна інформація містить список блоків LUT FPGA-проєкту, значення програмних кодів цих блоків і список зв'язків блоків LUT між собою.

2. Програмний додаток EsLUTGetter, який приймає від додатка LUTListExtractor інформацію про сукупність блоків LUT, їх зв'язки в FPGA-проєкті та виконує моделювання LUT-схеми з метою визначенні кількості суттєвих блоків LUT, що дає змогу оцінити кількість несуттєвих блоків LUT.

Для задач даної роботи було розроблено програмний додаток LMatrixLUTGetter, який пропонується долучити до набору додатків з локалізації несуттєвих блоків LUT (рис. 1). Цей додаток приймає від додатка LUTListExtractor інформацію про сукупність блоків LUT та їх зв'язки в FPGA-проєкті, аналізує структуру матриці обчислювача, визначає розмір молодшої частини матриці в структурі обчислювача та виконує пошук блоків LUT, які відносяться до молодшої частини матриці. За результатом функціонування додатка LMatrixLUTGetter отримується список несуттєвих блоків LUT, які розміщені в молодшій частині матриці обчислювача для кожного FPGA-проєкта. На основі аналізу цього списку виконується нижня оцінка кількості несуттєвих блоків LUT.

В середовищі вказаних програмних додатків було виконано експериментальне дослідження з локалізації несуттєвих блоків LUT в молодшої частині матриці обчислювача. Вихідні дані експерименту становили п'ять FPGA-проєктів, які складаються з модулів помноження мантис та мають розрядність кожного з операндів відповідно 4, 6, 8, 12 та 16 розрядів та розрядність до округлення відповідно 8, 12, 16, 24 та 32 розряди. Після округлення розрядність результату збігається з розрядністю операндів. Синтез, розміщення і трасування FPGA-проєктів виконувалися в САПР Intel Quartus Prime 20.1 Lite Edition для цільових мікросхем FPGA Intel Cyclone IV EP4CE15F23A7.

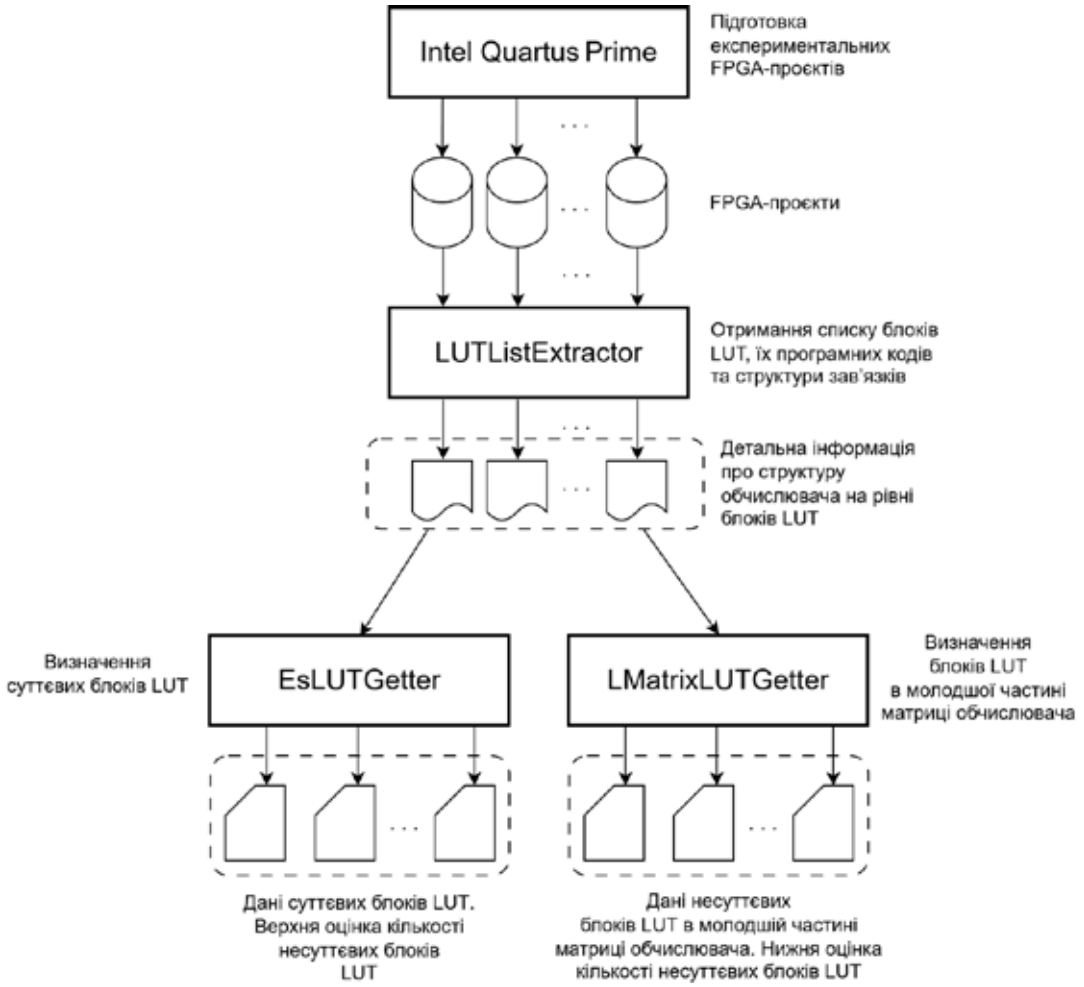

Рис. 1. Зв'язки між програмними додатками, які становлять середовище проведення експерименту з локалізації несуттєвих блоків LUT

Результати експериментального дослідження з визначення нижньої оцінки кількості несуттєвих блоків LUT та їх локалізації представлені в табл. 1. В експериментальних FPGA-прєектах нижня оцінка частки несуттєвих блоків LUT лежить в діапазоні 16,67% … 29,91% від загальної кількості блоків LUT проєктів. Обсяг, доступний для зберігання даних в стеганографічний спосіб, забезпечений несуттєвими блоками LUT, за отриманою нижньою оцінкою змінюється від 5 до 102 бітів та від 80 до 1632 бітів при використанні для вбудовування стего-даних відповідно тільки одного або всіх розрядів програмного коду несуттєвих блоків LUT. В ході проведення експериментального дослідження на обчислювальній системі на базі 8-ядерного процесора AMD Ryzen 7 при обсязі оперативної пам'яті 16 ГБ часова складність пошуку блоків LUT в молодшій частині матриці обчислювачів становила від $2.1 \cdot 10^{-5}$ с до $2.6 \cdot 10^{-4}$ с. В той час, як для цих же проєктів з застосуванням переборних методів в роботах [6, 7] час моделювання склав від $0.5 \cdot 10^{-4}$ с до 821.75 с.

Таблиця 1

Результати експериментального визначення нижньої оцінки кількості несуттєвих блоків LUT

Розрядність операндів та кінцевого результату	**4**	**6**	**8**	**12**	**16**
Розрядність результату до округлення	8	12	16	24	32
Загальні кількість блоків LUT	30	61	101	208	341
Нижня оцінка кількості несуттєвих блоків LUT	5	10	22	51	102
Нижня оцінка частки несуттєвих блоків LUT серед усіх блоків	16,67%	16,39%	21,78%	24,52%	29,91%
Мінімальний обсяг, доступний для зберігання даних (біт)	5	10	22	51	102
Максимальний обсяг (біт), доступний для зберігання даних при кількості входів блоків LUT – 4	80	160	352	816	1632

На комплексній діаграмі (рис. 2) з результатами експериментів з дослідження інформаційних стеганографічних ресурсів, забезпечених несуттєвими блоками LUT, показано порівняння абсолютних, а також відносних значень загальної кількості блоків LUT, кількість експерементально визначених в роботах [6, 7] суттєвих блоків LUT, верхньої та визначеної в даній роботі нижньої оцінки кількості несуттєвих блоків LUT для кожного з експериментальних FPGA-проєктів.

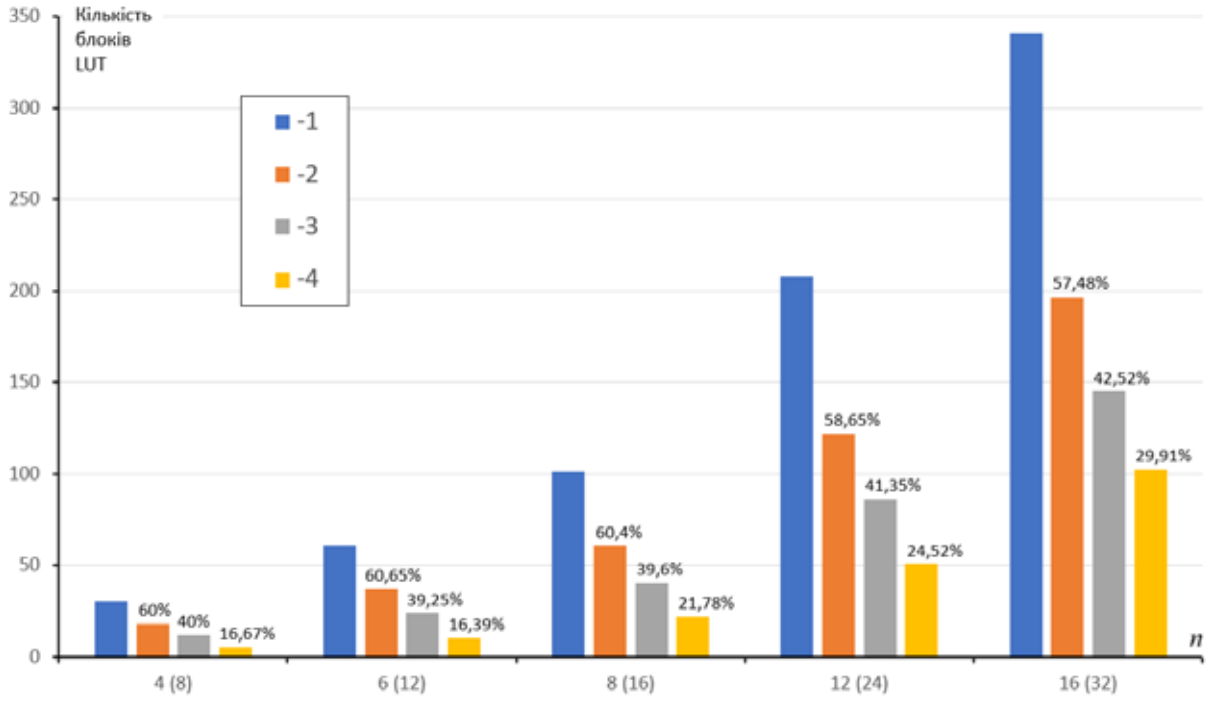

Рис. 2. Результати експериментального дослідження з урахуванням нижньої оцінки кількості несуттєвих блоків LUT

1 – загальна кількість блоків LUT в реалізації FPGA-проєкту; 2 – з них кількість експерементально визначених суттєвих блоків LUT; 3 – верхня оцінка кількості несуттєвих блоків LUT; 4 – нижня оцінка кількості несуттєвих блоків LUT

Висновки. В роботі запропоновано підхід швидкої локалізації несуттєвих блоків LUT в структурі обчислювачів, які виконують обробку чисел с плаваючою крапкою та реалізовані на елементній базі FPGA. На відміну

від відомого підходу до локалізації несуттєвих блоків LUT, підхід що запропоновано, має малу обчислювальну складність, але дає змого визначити тільки нижню оцінку кількості цільових блоків LUT та виконати їх локалізацію. В сукупності з відомими підходами, пропозиції даної роботи дають можливість обрати пошук повної множини несуттєвих блоків LUT при високій обчислювальній складності відомих підходів або пошук множини блоків, кількість елементів якої обмежено нижньою оцінкою кількості при низькій обчислювальній складності.

Виконане експериментальне дослідження показало, що навіть кількість несуттєвих блоків LUT, оцінена за нижньою оцінкою дає обсяг для прихованого збереження додаткових даних в програмному коді FPGA від 5 до 102 бітів при використанні для вбудовування стего-даних тільки одного розряду та від 80 до 1632 бітів при використанні для вбудовування стего-даних всіх розрядів програмного коду несуттєвих блоків LUT. Такі обсяги є співставними з розмірами хеш-сум та похідних кодів, що використовуються в якості контрольних в задачах моніторингу програмного коду. Зазначені обсяги суттєво доповнюють простір для прихованого зберігання контрольних даних та дають можливість забезпечити стеганографічне зберігання контрольних даних більшої кількості видів моніторингу.

Список використаної літератури

1. Tu K., Tang S., Yu X., Josipovic L., Chu Z. FPGA EDA: Design Principles and Implementation. Singapore: Springer, 2024. 248 p.

2. Ashutosh M., Goswami M., Manoj K., Rajput N. Hardware Security: Challenges and Solutions. Singapore: Springer, 2025. 274 p.

3. Shih F. Digital Watermarking and Steganography: Fundamentals and Techniques. 2nd ed. USA, Boca Raton: CRC Press. 2017. 320 p.

4. Drozd A., Antoshchuk S., Drozd J., Zashcholkin K., Drozd M., Kuznietsov N., Al-Dhabi M., Nikul V. Checkable FPGA Design: Energy Consumption, Throughput and Trustworthiness. *Studies in Systems, Decision and Control*: Green IT Engineering: Social, Business and Industrial Applications. Switzerland, Cham: Springer International Publishing, 2019. Vol. 171. P. 73-94.

5. Anderson A., Muralidharan S., Gregg D. Efficient Multibyte Floating Point Data Formats Using Vectorization, IEEE Transactions on Computers, vol. 66, no. 12 (2017) 2081-2096. DOI: 10.1109/TC.2017.2716355.

6. Zashcholkin K., Drozd O., Antoshchuk S., Ivanova O., Sachenko O. Steganographic Resources of FPGA-based Systems for Approximate Data Processing. *CEUR-WS*. 2021. Vol. 2864. P. 324-333.

7. Zashcholkin K., Drozd O., Ivanova O., Shaporin R., Kuznietsov M. An Approach to Stego-Container Organization in FPGA Systems for Approximate Data Processing. *CEUR-WS*. 2021. Vol. 2853. P. 527–536.

8. Дрозд О.В. Теоретичні основи, методи та засоби функціонального діагностування вузлів обчислювальних пристроїв з використанням природної надмірності при виконанні приблизних обчислень: дис. … д-ра техн. наук: 05.13.05. Одеса, 2003. 327 с.

References

1. Tu, K., Tang, S., Yu, X., Josipovic, L., Chu, Z. (2024). *FPGA EDA: Design principles and implementation*. Springer.

2. Ashutosh, M., Goswami, M., Manoj, K., Rajput, N. (2025). *Hardware security: Challenges and solutions*. Springer.

3. Shih, F. (2017). *Digital watermarking and steganography: Fundamentals and techniques* (2nd ed.). CRC Press.

4. Drozd, A., Antoshchuk, S., Drozd, J., Zashcholkin, K., Drozd, M., Kuznietsov, N., Al-Dhabi, M., Nikul, V. (2019). Checkable FPGA design: Energy consumption, throughput and trustworthiness. In *Studies in Systems, Decision and Control: Green IT Engineering: Social, Business and Industrial Applications* (Vol. 171, pp. 73–94). Springer International Publishing.

5. Anderson, A., Muralidharan, S., Gregg, D. (2017). Efficient multibyte floating point data formats using vectorization. *IEEE Transactions on Computers, 66*(12), 2081–2096. https://doi.org/10.1109/TC.2017.2716355

6. Zashcholkin, K., Drozd, O., Antoshchuk, S., Ivanova, O., Sachenko, O. (2021). Steganographic resources of FPGA-based systems for approximate data processing. *CEUR Workshop Proceedings*, 2864, 324–333.

7. Zashcholkin, K., Drozd, O., Ivanova, O., Shaporin, R., Kuznietsov, M. (2021). An approach to stego-container organization in FPGA systems for approximate data processing. *CEUR Workshop Proceedings*, 2853, 527–536.

8. Drozd, O.V. (2003). *Theoretic Bases, Methods and Means for On-line Testing of Computing Devices Units with Use of Natural Redundancy in Approximated Calculations Execution* (Doctoral dissertation).

Дата першого надходження рукопису до видання: 10.11.2025
Дата прийнятого до друку рукопису після рецензування: 08.12.2025
Дата публікації: 31.12.2025