

Д. В. Гаврілов<sup>1</sup>  
А. Ю. Воловик<sup>1</sup>  
О. С. Звягін<sup>1</sup>  
Д. В. Яровий<sup>1</sup>

## РЕВЕРСИВНИЙ ГЕНЕРАТОР КОДОВИХ ПОСЛІДОВНОСТЕЙ НА FPGA

<sup>1</sup>Вінницький національний технічний університет

*Генератори кодових послідовностей (ГКП) широко використовуються в цифрових системах радіотехніки та зв'язку, обчислювальної техніки та автоматики для зберігання інформації і виконання арифметичних операцій, а також діагностування і корекції похибок цифрових пристроїв у ланцюгах керування і синхронізації. Серед різного типу ГКП набули поширення генератори псевдовипадкових чисел і генератори зі сталими кодами, в яких сполучення нулів та одиниць у розрядах регістра залишається незмінним. Послідовність називається псевдовипадковою, якщо вона виглядає, як безсистемна і випадкова, хоча насправді вона створювалась за допомогою суто детермінованого процесу, відомого під назвою псевдовипадкового генератора. Подібні генератори переважно задаються деяким початковим значенням і за допомогою певних алгоритмів отримують з нього випадкові послідовності кодів. В цьому сенсі псевдовипадкові генератори можна розглядати як розповсюджувачі випадковості. До основних недоліків таких пристроїв слід віднести те, що виконання зсуву здійснюється тільки праворуч, що зменшує функціональні можливості.*

*Авторами проведено аналіз сучасних методів реалізації генераторів кодових послідовностей та, враховуючи їх недоліки, запропоновано нове схемне рішення реверсивного генератора кодових послідовностей, яке дозволяє зменшити використовуваний ресурс інтегральних схем, оскільки побудований на регістрі довільної розрядності, і таким чином дає можливість утворювати вихідну шину довільної розрядності без зміни вихідного коду в точці реверсу. Описано метод визначення функції збудження для  $n$ -розрядного регістра зсуву та наведено приклад розрахунку чотирирозрядного регістра зсуву, що забезпечує автоматичне повернення до основного режиму роботи системи. Таке рішення дозволяє створювати гнучкі системи на основі стандартних інтегральних мікросхем жорсткої логіки. Подано результати моделювання схеми реверсивного генератора кодових послідовностей у програмному забезпеченні САПР Altera Quartus II з наведенням часових діаграм роботи пристрою.*

**Ключові слова:** реверсивний генератор кодових послідовностей; метод визначення функції збудження; гнучкі системи стеження; інтегральні мікросхеми жорсткої логіки; Altera; Quartus; Intel.

### Вступ

У статистичному моделюванні виникає необхідність в генерації спочатку прямої, а потім зворотної послідовностей випадкових (або псевдовипадкових) рівномірно розподілених двійкових чисел, що характеризують деякі проміжні стани модельованого процесу [1]. Для цього використовують реверсивні генератори кодових послідовностей (РГКП), утворені на основі регістрів зсуву і коригувальних цифрових комбінаційних пристроїв (КЦКП). Але такі рішення мають недоліки: через складні міжрозрядні зв'язки реверсивні регістри зсуву потребують багато ресурсу ІС та втрачають гнучкість керування (необхідно перемикає входи та виходи послідовного введення/виведення, ускладнюється реалізація РГКП довільної розрядності) [2]—[5].

*Метою роботи є розробка  $n$ -розрядного реверсивного генератора кодових послідовностей з довільною вихідною шиною розрядності без зміни вихідного коду в точці реверсу та зменшення потрібного ресурсу інтегральних схем, а також створення гнучких систем на основі стандартних інтегральних мікросхем жорсткої логіки.*

## Результати дослідження

Для досягнення поставленої мети в РГКП використано регістр зсуву довільної розрядності з входами і виходами паралельного введення/виведення, до якого додається мультиплексор шин, пристрій керування та КЦКП, реалізація схеми проводилась у відповідності до методик, описаних у [6], [7], схема показана на рис. 1.

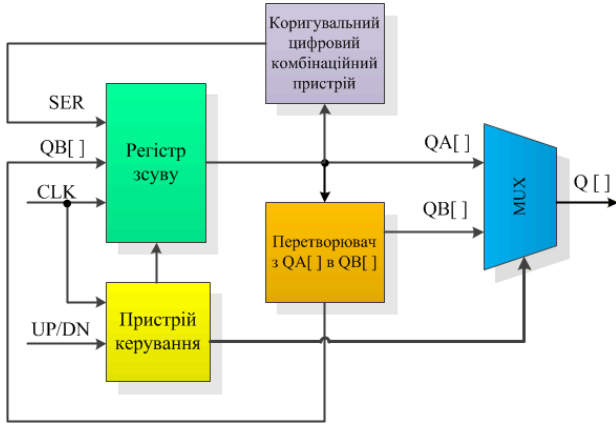


Рис. 1. Реверсивний генератор кодових послідовностей

Принцип роботи схеми такий: шинні входи мультиплексора шин з'єднані з виходами регістра зсуву двома шинами: одна шина QA[ ] утворюється з потрібної кількості розрядів регістра зсуву, починаючи з молодшого, в прямому порядку їх нумерації, а друга шина QB[ ] — з тих самих розрядів у зворотному порядку їх нумерації, причому друга шина з'єднується також зі входами паралельного введення регістра зсуву. Із синхросигналу CLK, що одночасно надходить до регістра зсуву і пристрою керування, та сигналу керування напрямком зсуву UP/DN, який одночасно є адресним входом мультиплексора шин, у пристрої керування формуються сигнали керування, якими здійснюється запис зворотного коду QB[ ] під час зміни напрямку зсуву, наприклад, два сигнали: сигнал, що подається на вхід регістра зсуву MODE, тобто перемикає регістр з послідовного в паралельний і навпаки, та імпульс завантаження, що подається на інверсний вхід CLKL. Вхід КЦКП з'єднаний з першою шиною даних з відповідними розрядами, а вихід з'єднаний з послідовним входом введення SER регістра зсуву. Вихід RESULT мультиплексора шин є виходом РГКП Q[ ].

Під час роботи пристрою може виникнути збій (наприклад внаслідок зовнішніх чинників), що призведе до зациклення системи. Відновити нормальну роботу можна двома шляхами: коли регістр потраплятиме до хибного циклу, примусово встановити один зі станів робочого циклу (наприклад, встановити початкове значення системи) або скоригувати функцію збудження таким чином, щоб хибні цикли не утворювалися.

Основою КЦКП є функція збудження, яка забезпечує повернення до основного робочого циклу. Для прикладу візьмемо послідовність символів з періодом (00001111).

Потрібно вибрати розрядність регістра зсуву так, щоби з проходженням послідовності утворювались детерміновані коди, тобто без повторення. В цьому випадку підходить чотирирозрядний регістр зсуву  $n = 4$ . Так, у разі проходження послідовності утворюються коди

$$N = Q2Q1Q1Q0 = (00001111) = 0, 1, 3, 7, 15, 14, 12, 8.$$

Визначаємо модуль  $M = 8$ , що дорівнює кількості символів у послідовності. Під час зсуву числа з послідовності символів на один крок у бік старших розрядів значення початкового коду  $N = Q2Q1Q1Q0$  подвоюється (наприклад,  $001 \rightarrow 010$ ) і до молодшого розряду додається ще наступний символ  $x_N$  послідовності, тобто з надходженням чергового синхроімпульсу матимемо код  $N^+ = 2N + x_N$ . Але якщо є одиниця в старшому розряді, то у наступному такті зсуву вона зникає, тому її вагу  $2^{n-1}$  перед подвоєнням потрібно відняти. Отже, наступний код можна визначити з виразів

$$N^+ = 2N + x_N, \text{ якщо } N < 2^{n-1}. \quad (1)$$

$$N^+ = 2(N - 2^{n-1}) + x_N, \text{ якщо } N > 2^{n-1}. \quad (2)$$

У чотирирозрядному регістрі зсуву  $n = 4$  перший код утворюється чотирма нулями послідовності символів (00001111), тобто  $N = Q2Q1Q1Q0 = 0$ . Другий код  $N^+ = 2 \times 0 + 1 = 1$ , оскільки  $N < 8$  і наступний символ послідовності  $x_N = 1$ ; так само третій код становить  $N^+ = 2 \times 1 + 1 = 3$  і так до останнього символу послідовності. По завершенні останнього символу з періоду послідовності переходимо до його початку. У підсумку отриманий результат нанесемо на

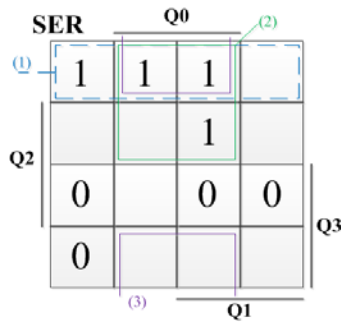


Рис. 2. Карта Карно для послідовного введення регістра зсуву

карту Карно для послідовного входу введення *SER* регістра зсуву.

На карті Карно (рис. 2) показано 8 порожніх комірок, з невизначеним станом, тобто вони можуть набувати значень як 1 так і 0, в результаті отримаємо  $2^8$  (256) можливих комбінацій для порожніх комірок.

Постає завдання створити функцію збудження так, щоб виконувались такі критерії:

- мінімальна кількість об’єднань;
- мінімальне значення коефіцієнта складності функції збудження, яке визначається як відношення  $N_{\text{елм}}$  кількості елементів на суму всіх входів  $\sum Bx$  елементів  $q = N_{\text{елм}} / \sum Bx$ ;
- у разі утворення хибних комбінацій повернення до основного циклу роботи відбувалось за найменшу кількість тактів, яка не перевищує половини тактів основної послідовності (у цьому прикладі  $M/2 = 4$ ).

Реверсивні генератори кодових послідовностей реалізовано у програмному забезпеченні САПР Altera Quartus II, схема такого регістра показана на рис. 3.

Реверсивні генератори кодових послідовностей апаратно реалізовано на ПЛІС родини FLEX10K Altera EPF10K70RC240-4. Мультиплексор шин реалізується на мегафункції IC програмованої структури або з’єднанням мультиплексорів IC жорсткої структури [8]—[11].

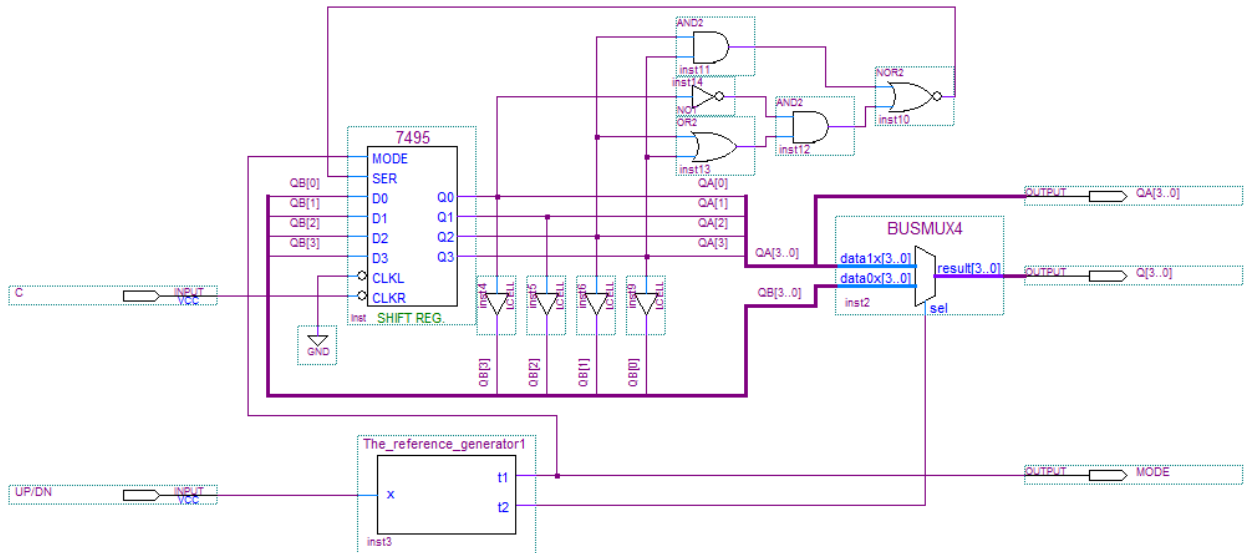


Рис. 3. Схема універсального регістра у САПР Quartus II

На рис. 3 пристрій керування подано у вигляді блока, який складається з декількох міжкоміркових буферів ПЛІС, з’єднаних послідовно між собою, причому кількість буферів залежить від затримки, яка потрібна для забезпечення роботи регістра зсуву. На виході  $t_1$  пристрою керування формується сигнал з відповідною шириною для запису зворотного коду, а на  $t_2$  відбувається затримка керувального сигналу, що надходить на адресний вхід мультиплексора шин для узгодження затримки між сигналом  $t_1$ . Виходи регістра, що утворюють першу шину даних, приєднані до другої шини даних через буфери для узгодження затримки з сигналом, що утворюється під час перемикавання напрямку зсуву.

Пристрій керування можна охарактеризувати як опорний генератор, що формує імпульси з певною частотою, яка залежить від керувального сигналу під час перепадів керувального сигналу UP/DN.

Для отримання мінімальної функції збудження потрібно виконати одне об’єднання верхньої частини карти Карно з одиницями, в результаті отримаємо функцію збудження:

$$SER = \overline{Q_3}. \tag{3}$$

Результати моделювання отриманої функції збудження за допомогою програмного забезпечення

САПР Altera Quartus II та експериментальне випробування у відповідності до [12] показано на рис. 4.

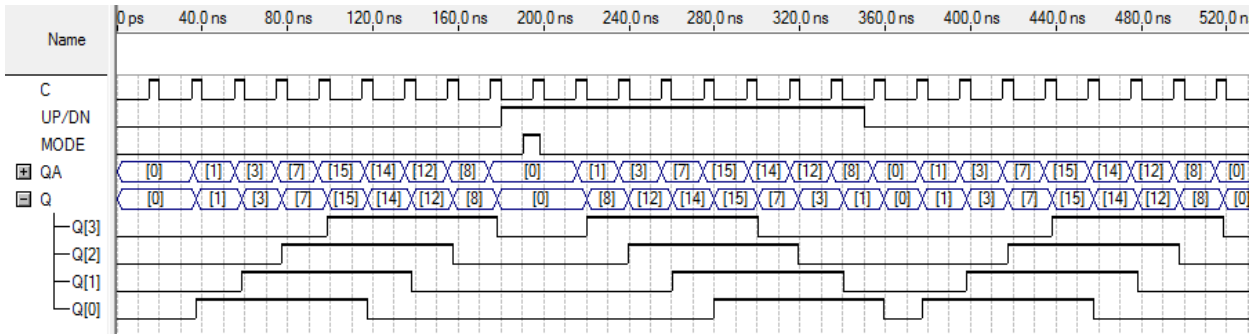


Рис. 4. Часові діаграми роботи пристрою

Як бачимо на рис. 4, функція збудження (3) забезпечує перехід між станами основної кодової послідовності в двох напрямках зсуву. Розглянемо випадок, коли на входи регістра зсуву випадково потрапляють хибні комбінації, таким чином визначимо чи отримана функція збудження здатна повернутись до основної послідовності. Для перевірки додатково на паралельні входи приєднано до регістра зсуву шину даних Qp[ ] та MODE1, щоб завантажити хибні значення коду, які не входять до основного робочого циклу (рис. 5).

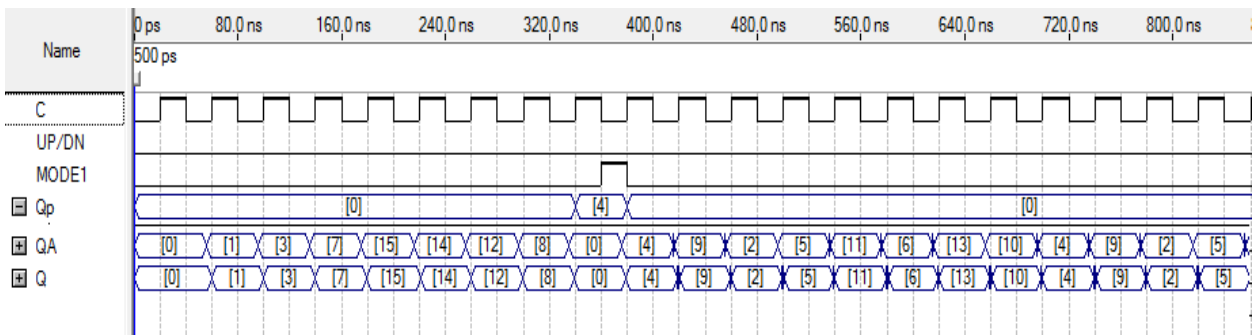


Рис. 5. Часові діаграма перевірки функції збудження (3)

В результаті перевірки функції збудження (див. рис. 5) виявлено, що з попаданням хибного коду до регістра зсуву відбувається зациклення системи, причому утворюється хибний цикл, який складається зі всіх можливих комбінацій, які не входять до основної кодової послідовності.

Для формування функції збудження, яка б забезпечувала повернення до основної послідовності у випадку хибної комбінації з дотриманням поставлених критерій, проведено аналіз можливих переходів для кожної невизначеної комірки, за допомогою формул (1) і (2), при попаданні як 1, так і значення 0. В результаті аналізу виявлено, що найкращим варіантом для цього прикладу є три об'єднання, як показано на карті Карно

$$SER = \overline{Q_2} \overline{Q_3} + Q_0 \overline{Q_3} + Q_0 \overline{Q_2} \rightarrow \overline{Q_2} \overline{Q_3} + (Q_2 + Q_3) \overline{Q_0}. \quad (4)$$

Отриману функцію збудження за допомогою моделювання перевіримо на самовідновлення, завантаживши на паралельні входи регістра зсуву всі можливі хибні комбінації у разі зсувів як ліворуч, так і праворуч.

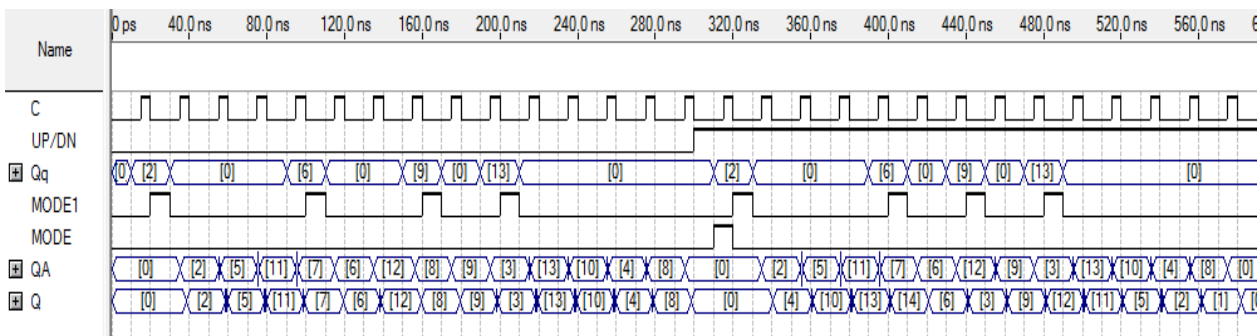


Рис. 6. Часова діаграма сформованої функції збудження

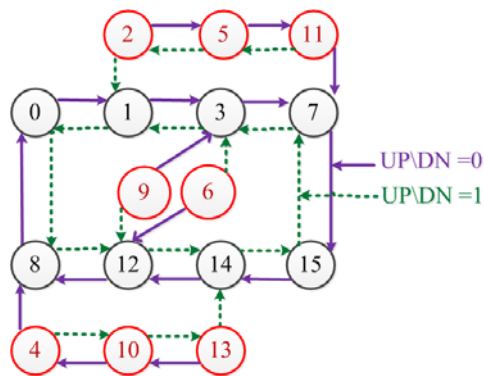


Рис. 7. Повний перемикальний граф

За результатом моделювання (рис. 6) складемо перемикальний граф для кращої візуалізації переходів між станами роботи системи, показаний на рис. 7. На перемикальному графі показано зсув кодової послідовності праворуч  $UP/DN=0$  та ліворуч  $UP/DN=1$  до повторення; так умовно графік можна розділити на два частини, в першій частині перехід між станами основної послідовності, а в другій перехід між хибними станами до основного режиму роботи системи.

Отримана функція збудження для чотирирозрядного РГКП в цілому відповідає встановленим критеріям, оскільки містить тільки 3 об'єднання, коефіцієнт складності функції збудження становить

$$q = N_{\text{елм}} / \sum Bx = \frac{5}{9}. \quad (5)$$

Максимальний період повернення від хибної комбінації до основної кодової послідовності відбувається за три такти в двох напрямках зсуву. У разі потреби налагоджувальна інформація може відобразитись на семисегментному індикаторі за допомогою спеціалізованого дешифратора [13].

### Висновки

Проведено аналіз сучасних методів формування генераторів кодових послідовностей та враховуючи їх недоліки авторами запропоновано нове схемне рішення реверсивного генератора кодових послідовностей, що дозволяє зменшити використання потрібного ресурсу ІС, оскільки побудований на регістрі довільної розрядності  $i$ , таким чином, дає можливість утворювати вихідну шину довільної розрядності без зміни вихідного коду в точці реверсу. Визначено метод формування функції збудження для  $n$ -розрядного регістра зсуву, що дає можливість працювати в автоматичну режимі.

Керування реверсивним генератором кодових послідовностей виконується лише одним сигналом  $UP/DN$ , яким не тільки можна змінювати напрямок зсуву кодової послідовності, а також періодом повторення кодових послідовностей за рахунок тривалості керувального сигналу.

### СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] Q. Wang, and S. Yu, C. Guyeux, Design of Digital Chaotic Systems Updated by Random Iterations (SpringerBriefs in Applied Sciences and Technology), Springer, 1st ed. 2018 ed., 124 p., 2018. ISBN: 978-3319735481.
- [2] И. Н. Букреев, В. И. Горячев, и Б. М. Мансуров, *Микроэлектронные схемы цифровых устройств* 4-е изд., перераб. и доп. Москва, Россия: Техносфера, 2009.
- [3] Chaofan Yu, Lingli Wang and Xuegong Zhou, "Automatic layout generator for embedded FPGA cores," in *2011 9th IEEE International Conference on ASIC*, Xiamen, 2011, pp. 385-388. <https://doi.org/10.1109/ASICON.2011.6157202>.
- [4] P. Pfeifer, F. Hosseinzadeh and H. T. Vierhaus, "On comparison of configurable encoders in Xilinx and Altera FPGAs," *2017 International Conference on Applied Electronics (AE)*, Pilsen, 2017, pp. 1-4. <https://doi.org/10.23919/AE.2017.8053605>.
- [5] Y. Ruan, Y. Tang, W. Yao, Z. Wang and S. Xu, "A single chip multi-functional DDS waveform generator based on FPGA with SOPC design flow," *Proceedings of the 10th World Congress on Intelligent Control and Automation*, Beijing, 2012, pp. 4206-4210. <https://doi.org/10.1109/WCICA.2012.6359183>.
- [6] Д. В. Гаврілов, О. В. Осадчук, та О. С. Звягін, *Основи комп'ютерного проектування та моделювання РЕА. Частина 1, лабораторний практикум*. Вінниця, Україна: ВНТУ, 2015.
- [7] Д. В. Гаврілов, О. С. Звягін, О. В. Осадчук, та А. Ю. Савицький. *Основи комп'ютерного проектування та моделювання РЕА. Частина 2, лабораторний практикум*. Вінниця, Україна: ВНТУ, 2017.
- [8] В. Л. Кофанов, О. В. Осадчук, Д. В. Гаврілов. *Лабораторний практикум з цифрових пристроїв на основі САПР Quartus II*. Вінниця, Україна: УНІВЕРСУМ, 2007.
- [9] Cetin Kaуа Кос, Cryptographic Engineering, Springer, 2009 ed., 522 p., 2009. ISBN: 978-0387718163.
- [10] В. Л. Кофанов, О. В. Осадчук, та Д. В. Гаврілов. *Лабораторний практикум з дослідження цифрових пристроїв на основі САПР MAX+PLUS II*. Вінниця, Україна: УНІВЕРСУМ, 2006.
- [11] Esteban Tlelo-Cuautle, José de Jesús Rangel-Magdaleno, and Luis Gerardo de la Fraga, *Engineering Applications of FPGAs: Chaotic Systems, Artificial Neural Networks, Random Number Generators, and Secure Communication Systems*, Springer, 1st ed., 222 p., 2016, ISBN: 978-3319341132.
- [12] А. Ю. Воловик, Д. В. Гаврілов, А. О. Семенов, М. А. Шутило, та О. П. Червак. *Сигнали та процеси в радіотехніці: лабораторний практикум*. Вінниця, Україна: ВНТУ, 2015.

[13] Б. В. Басич, Д. В. Гаврілов, О. Е. Белов, та Ю. В. Ситай, «Дешифратор семисегментного коду в унітарний,» *МКИ Н03М 13/00, Н03М 7/00. № u201600023*, Липень 25, 2016.

Рекомендована кафедрою радіотехніки ВНТУ

Стаття надійшла до редакції 29.05.2019

**Гаврілов Дмитро Володимирович** — канд. техн. наук, доцент, доцент кафедри радіотехніки, e-mail: havrilov@vntu.edu.ua ;

**Воловик Андрій Юрійович** — канд. техн. наук, доцент кафедри радіотехніки, e-mail: voland@vntu.edu.ua ;

**Звягін Олександр Сергійович** — канд. техн. наук, доцент кафедри радіотехніки, e-mail: zviahin86@gmail.com ;

**Яровий Дмитро Володимирович** — студент факультету інфокомунікацій, радіоелектроніки та наносистем, email: rt15b.yarovoy@gmail.com

**D. V. Havrilov<sup>1</sup>**  
**A. Yu. Volovyk<sup>1</sup>**  
**O. S. Zviahin<sup>1</sup>**  
**D. V. Yarovy<sup>1</sup>**

## Reversible FPGA-Based Code Sequence Generator

<sup>1</sup>Vinnitsia National Technical University

*Code sequence generators (CSGs) are widely used in digital systems of radio engineering and communication, computer technology and automation, for storing information and performing arithmetic operations, as well as diagnosing and correcting digital device errors in their control and synchronization circles. Among various types of CSG propagation generators were pseudo-random numbers and generators with constant codes, in which the combination of zeros and units in the bits of the register remains unchanged. A sequence is called pseudorandom, if it looks like a systemic and random, although in fact it was created using a purely deterministic process known as the pseudo-random generator. Such generators are predominantly given some initial values and, with the help of certain algorithms, receive random sequences of codes from it. In this sense, pseudorandom generators can be considered as spreaders of chance. The main disadvantages of these devices include the fact that the offset is performed only to the right, which reduces the functionality.*

*In this article, an analysis of modern methods for generating code sequence generators and taking into account their shortcomings, the authors proposed a new schematic solution for a reversible code sequence generator, which reduces the required resource of integrated circuits, so that it is constructed on an arbitrary bit register, thus giving an opportunity to form an output a bus of any size without changing the source code at the reverse. The method of determining the excitation function for the n-bit shift register is described, and the example of calculating the 4-bit shift register is provided, which provides a return to the main work of the system. This solution allows creating flexible systems based on standard integrated circuits of hard logic. The result of the simulation of the reverse code sequence generator in the software of the Altera Quartus II CAD with the time charts of the device operation is described.*

**Keywords:** reverse code sequence generator; excitation function determination method; flexible tracking system; integral chips of hard logic; Altera; Quartus; Intel.

**Havrilov Dmytro V.** — Cand. Sc. (Eng), Associate Professor, Associate Professor of the Chair of Radio Engineering, e-mail: havrilov@vntu.edu.ua ;

**Volovyk Andrii Yu.** — Cand. Sc. (Eng), Associate Professor of the Chair of Radio Engineering, e-mail: voland@vntu.edu.ua ;

**Zviahin Oleksandr S.** — Cand. Sc. (Eng.), Associate Professor of the Chair of Radio Engineering, e-mail: zviahin86@gmail.com ;

**Yarovy Dmytro V.** — Student of the Department of Infocommunications, Radioelectronics and Nanosystems, email: rt15b.yarovoy@gmail.com

Д. В. Гаврилов<sup>1</sup>  
А. Ю. Воловик<sup>1</sup>  
А. С. Звягин<sup>1</sup>  
Д. В. Яровой<sup>1</sup>

## Реверсивный генератор кодовой последовательности на FPGA

<sup>1</sup>Вінницький національний технічний університет

Генераторы кодовых последовательностей (ГКП) широко используются в цифровых системах радиотехники и связи, вычислительной техники и автоматики, для хранения информации и выполнения арифметических операций, а также диагностирования и коррекции ошибок цифровых устройств, в цепях управления и синхронизации. Среди различного типа ГКП получили распространение генераторы псевдослучайных чисел и генераторы с постоянными кодами, в которых сочетание нулей и единиц в разрядах регистра остается неизменным. Последовательность называется псевдослучайной, если она выглядит, как бессистемная и случайная, хотя на самом деле она создавалась с помощью детерминированного процесса, известного под названием псевдослучайного генератора. Подобные генераторы преимущественно задаются некоторым начальным значением и с помощью определенных алгоритмов получают из него случайные последовательности кодов. В этом смысле псевдослучайные генераторы можно рассматривать как распространители случайности. К основным недостаткам данных устройств, следует отнести то, что выполнение смещения осуществляется только вправо, что уменьшает функциональные возможности.

Проведен анализ современных методов формирования генераторов кодовых последовательностей и, учитывая их недостатки, авторами предложено новое схемное решение реверсивного генератора кодовых последовательностей, что позволяет уменьшить необходимый ресурс интегральных схем, поскольку реализован на регистре произвольной разрядности, таким образом, это дает возможность создавать выходную шину произвольной разрядности без изменения выходного кода в точке реверса. Описан метод определения функции возбуждения для  $n$ -разрядного сдвигового регистра и приведен пример расчета четырехразрядного кода, который обеспечивает автоматический возврат к основной работе системы. Такое решение позволяет создавать гибкие системы на основе стандартных интегральных микросхем жесткой логики. Изложены результаты моделирования схемы реверсивного генератора кодовых последовательностей в программном обеспечении САПР Altera Quartus II с временными диаграммами работы устройства.

**Ключевые слова:** реверсивный генератор кодовых последовательностей; метод определения функции возбуждения; гибкие системы слежения; интегральные микросхемы жесткой логики; Altera; Quartus; Intel.

**Гаврилов Дмитрий Владимирович** — канд. техн. наук, доцент, доцент кафедры радиотехники, email: havrilov@vntu.edu.ua ;

**Воловик Андрей Юрьевич** — канд. техн. наук, доцент кафедры радиотехники, e-mail: voland@vntu.edu.ua ;

**Звягин Александр Сергеевич** — канд. техн. наук, доцент кафедры радиотехники, e-mail: zviahin86@gmail.com ;

**Яровой Дмитрий Владимирович** — студент факультета инфокоммуникаций, радиоэлектроники и наносистем, e-mail: rt15b.yarovoy@gmail.com