ЗАЩИТА ИНФОРМАЦИИ И НАДЕЖНОСТЬ СИСТЕМ
INFORMATION PROTECTION AND SYSTEM RELIABILITY

УДК 004.832.32
https://doi.org/10.37661/1816-0301-2024-21-1-9-27

Оригинальная статья
Original Paper

Симметричные физически неклонируемые функции типа арбитр

В. Н. Ярмолик✉, А. А. Иванюк

Белорусский государственный университет информатики и радиоэлектроники,
ул. П. Бровки, 6, Минск, 220013, Беларусь
✉*E-mail: yarmolik10ru@yahoo.com*

Аннотация

Цели. Решается задача построения нового класса физически неклонируемых функций типа арбитр (АФНФ), объединяющих достоинства как классических, так и сбалансированных АФНФ. Актуальность такого исследования связана с активным развитием физической криптографии. В работе преследуются следующие цели: исследование и анализ классических АФНФ, построение новой математической модели АФНФ и разработка нового базового элемента АФНФ.

Методы. Используются методы синтеза и анализа цифровых устройств, в том числе на программируемых логических интегральных схемах, основы булевой алгебры и схемотехники.

Результаты. Установлено, что в классических АФНФ применяется стандартный базовый элемент, выполняющий три функции, а именно функцию формирования двух случайных величин *Generate*, функцию выбора пары путей *Select* и функцию переключения путей *Switch*, которые задаются одним битом запроса. Показано, что совместное использование этих функций, с одной стороны, позволяет достичь высоких характеристик АФНФ, а с другой – приводит к формированию асимметричного поведения АФНФ. С целью анализа основных характеристик АФНФ и их идеального поведения была рассмотрена новая математическая модель АФНФ, аналогичная модели случайного подбрасывания монеты. Для реализации АФНФ, функционирующих согласно предложенной модели, был разработан новый базовый элемент. Показано, что применение предложенного базового элемента позволяет строить симметричные физически неклонируемые функции (С_АФНФ), отличающиеся от классических АФНФ тем, что функции *Generate*, *Select* и *Switch* базового элемента выполняются независимыми его компонентами и задаются разными битами запроса.

Заключение. Предложенный подход к построению симметричных физически неклонируемых функций, основанный на реализации функций *Generate*, *Select* и *Switch* различными компонентами базового элемента, показал свои работоспособность и перспективность. Экспериментально подтвержден эффект улучшения характеристик подобных С_АФНФ, и в первую очередь заметного улучшения их вероятностных свойств, выраженных в равной вероятности ответов. Перспективным представляется дальнейшее развитие идей построения С_АФНФ, экспериментальное исследование их характеристик, а также анализ устойчивости к различного рода атакам, в том числе и с использованием машинного обучения.

Ключевые слова: физическая криптография, физически неклонируемые функции, физические однонаправленные функции, физически неклонируемая функция типа арбитр

Для цитирования. Ярмолик, В. Н. Симметричные физически неклонируемые функции типа арбитр / В. Н. Ярмолик, А. А. Иванюк // Информатика. – 2024. − Т. 21, № 1. – С. 9–27.
https://doi.org/10.37661/1816-0301-2024-21-1-9-27

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 16.07.2023
Подписана в печать | Accepted 03.01.2024
Опубликована | Published 29.03.2024

Symmetric physically unclonable functions of the arbiter type

Vyacheslav N. Yarmolik✉, Alexander A. Ivaniuk

Belarusian State University of Informatics and Radioelectronics,
st. P. Brovki, 6, Minsk, 220013, Belarus
✉*E-mail: yarmolik10ru@yahoo.com*

Abstract
Objectives. The problem of constructing a new class of physically unclonable functions of the arbiter type (APUF) that combines the advantages of both classical and balanced APUF is solved. The relevance of such a study is associated with the active development of physical cryptography. The following goals are pursued in the work: research and analysis of classical APUF, construction of a new mathematical model of APUF and development of a new basic element of APUF.
Methods. The methods of synthesis and analysis of digital devices are used, including those based on programmable logic integrated circuits, the basics of Boolean algebra and circuitry.
Results. It has been established that classical APUF uses a standard basic element that performs three functions, namely, the function of generating two random variables *Generate*, the function of choosing a pair of paths *Select* and the function of switching paths *Switch*, which are specified by one bit of the challenge. It is shown that the joint use of these functions, on the one hand, makes it possible to achieve high characteristics of the APUF, and on the other hand, leads to the formation of an asymmetric behavior of the APUF. In order to analyze the main characteristics of APUF and their ideal behavior, a new mathematical model of APUF was considered, similar to the model of random coin toss. To implement APUF functioning according to the proposed model, a new basic element was developed. It is shown that the use of the proposed basic element allows to build symmetrical physically unclonable functions (C_APUF), which differ from the classical APUF in that the *Generate*, *Select* and *Switch* functions of the basic element are performed by their independent components and are specified by different bits of challenge.
Conclusion. The proposed approach to the construction of symmetrical physically unclonable functions, based on the implementation of the *Generate*, *Select* and *Switch* functions by various components of the base element, has shown its efficiency and promise. The effect of improving the characteristics of similar C_APUF has been experimentally confirmed, and, first of all, a noticeable improvement in their probabilistic properties expressed in equal probability of responses. It seems promising to further develop the ideas of building C_APUF, experimental study of their characteristics, as well as analysis of resistance to various types of attacks, including using machine learning.

Keywords: physical cryptography, physically unclonable functions, physical one-way functions, physically unclonable arbiter-type function

For citation. Yarmolik V. N., Ivaniuk A. A. *Symmetric physically unclonable functions of the arbiter type*. Informatika [*Informatics*], 2024, vol. 21, no. 1, pp. 9−27 (In Russ.).
https://doi.org/10.37661/1816-0301-2024-21-1-9-27

Conflict of interest. The authors declare of no conflict of interest.

Введение. В настоящее время многие цифровые устройства, связанные с нашей повседневной жизнью, подключены к вычислительным системам и сетям, что требует решения задач их идентификации и аутентификации. Физически неклонируемые функции (ФНФ) (Physical Unclonable Functions, *PUF*s) [1, 2] были предложены для решения этих задач [3]. В последние годы сфера применения PUF значительно расширилась за счет их активного использования в криптографии для генерирования криптографических ключей, а также реализации различных криптографических протоколов [4, 5].

Наиболее широко используемое на сегодняшний день определение ФНФ было предложено П. Туилсом (P. Tuyls) [3]. Согласно его формулировке физически неклонируемые функции – это физические системы, неотъемлемым свойством которых является неклонируемость, т. е. невозможность воспроизведения двух идентичных ФНФ. Подобные системы наследуют данное свойство неклонируемости из-за того, что состоят из множества компонентов, параметры которых в процессе создания подобных физических систем принимают случайные значения [3, 6–8]. Невозможность контролировать и управлять параметрами элементов ФНФ, принимающими случайные значения во время производства, делает их уникальными и физически неклонируемыми. ФНФ описываются значениями входных и выходных параметров (сигналов). Подобная пара, состоящая из входного физического параметра запроса (Challenge – *C*) и выходного параметра ответа (Response – *R*), называется парой запрос-ответ (Challenge-Response Pair, *CRP*). Таким образом, ФНФ можно рассматривать как функцию $R = F(C)$, которая преобразует запросы *C* в ответы *R* [3, 6–8].

Анализ большого числа исследований в области ФНФ [3, 6–12] показывает, что в общем случае из всех характеристик, описывающих поведение ФНФ, на первом месте стоит стабильность, характеризующаяся повторяемостью ответов на один и тот же запрос. Затем идет уникальность, которая напрямую связана с неклонируемостью ФНФ, далее отмечают простоту технической реализации и, наконец, непредсказуемость, описывающую случайность ФНФ. Наиболее полно всем приведенным характеристикам отвечают ФНФ, основанные на задержках распространения (delay based) электрических сигналов [6, 8–15].

Существует множество разнообразных реализаций ФНФ, использующих задержки распространения тестового сигнала, среди которых лидирующую позицию занимают так называемые АФНФ (*APUF*) [6, 16–19]. На основании запроса *C* в АФНФ задается конфигурация, как правило, двух функционально и топологически симметричных путей, по которым распространяются идентичные копии тестового сигнала. Ответом *R* АФНФ является результат сравнения временных задержек распространения сигнала по двум путям [6]. Симметричность путей, определяющая их идентичность, обеспечивает близкие значения величин задержек распространения сигналов, которые в силу технологических вариаций в процессе производства, имеющих случайный характер, будут иметь незначительные отличия. Пары симметричных путей АФНФ изготавливаются таким образом, чтобы подобных пар было большое множество, из которого по конкретному запросу *C* выбирается одна из них.

Классической схемой АФНФ является схема, изображенная на рис. 1 [6, 8]. Она строится с использованием *n* последовательно подключенных базовых элементов, состоящих из пар двухвходовых мультиплексоров MUX_1 и MUX_2.

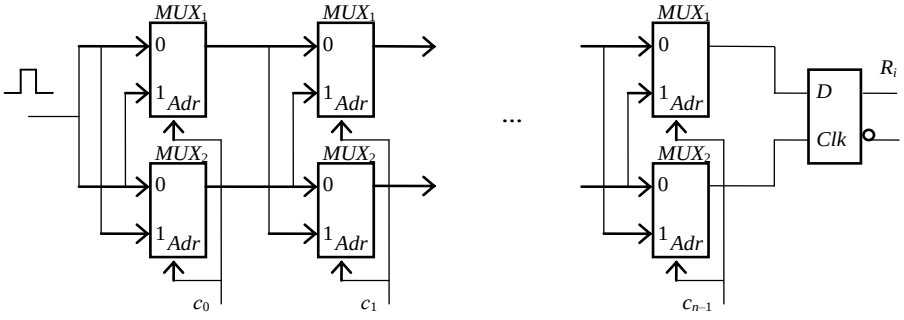


Рис. 1. ФНФ типа арбитр на базе двухвходовых мультиплексоров

Fig. 1. Arbiter-type PUF based on two-input multiplexers

Управляющие входы (*Adr*) мультиплексоров MUX_1 и MUX_2 каждой пары являются одним из входов для задания значения бита c_j запроса C_i, представляющего собой n-разрядный двоичный вектор $C_i = c_0 c_1 c_2 ... c_{n-1}$, где $c_j \in \{0, 1\}$, $j \in \{0, 1, 2, \ldots, n-1\}$. Запрос C_i в схеме АФНФ (см. рис. 1) формирует два пути таким образом, что если для j-й ступени АФНФ $c_j = 0$, то для построения первого пути используется мультиплексор MUX_1, для второго пути MUX_2, а если $c_j = 1$ – наоборот.

Основные проблемы при создании ФНФ заключаются в противоречии требования, которое характеризует стабильность их функционирования, с требованием о непредсказуемости, случайности таких функций. Попытка увеличить стабильность ФНФ увеличивает их уязвимость для различного рода атак, в особенности атак с применением современных методов машинного обучения [20, 21].

Случайность АФНФ оценивается метрикой единообразия (uniformity), которая определяет равновероятность появления ответов 0 и 1. Значения данной метрики меньше 1,0 свидетельствуют о наличии асимметрии в генерируемых парах путей, что особенно характерно для АФНФ, реализованных на программируемой логике (FPGA) [15, 16, 19, 22]. Одним из наиболее эффективных методов увеличения симметрии пар путей АФНФ является их балансировка [22–24]. Эта процедура, требующая дополнительных индивидуальных настроек АФНФ, технологически является сложной задачей, а в ряде случаев ASIC-реализаций и невыполнимой.

Как показано в работах [22, 24], основная причина асимметрии – это взаимозависимость трех функций, реализуемых базовым элементом одновременно. В общем случае задача балансировки пар путей АФНФ решается путем модификации классического базового элемента [22–24]. В статье [24] приведен оригинальный базовый элемент, позволяющий нивелировать асимметрию за счет реализации функции генерирования случайной величины добавленной задержки не на мультиплексорах, а на выделенных линиях задержки. Однако мультиплексоры базового элемента по-прежнему одновременно реализуют две остальные функции, связанные с выбором добавленной задержки и формированием пары путей. По сути, базовый элемент, рассмотренный в работе [24], определяет только знак добавленной задержки, который однозначно влияет на выбор одного из двух путей через базовый элемент. Вторым существенным недостатком базового элемента, приведенного в [24], является необходимость балансировки запросов $C_i = c_0 c_1 c_2 ... c_{n-1}$, $c_j \in \{0, 1\}$. Балансировка заключается в использовании только таких запросов, для которых выполняется баланс единичных и нулевых значений c_j. Соответственно, n должно быть четным, а количество и вид возможных значений запросов C_i ограниченными.

Таким образом, проблема построения эффективных АФНФ, характеризующихся высокой степенью симметрии, как наиболее распространенной разновидности ФНФ является практически нерешенной. В предложенной статье рассматривается задача построения симметричных АФНФ, которые характеризуются обеспечением высоких показателей их характеристик, таких как стабильность, уникальность и единообразие. Материал данной статьи является дальнейшим развитием идей балансировки пар путей АФНФ, изложенных в работе [24]. В сравнении с ранее полученными результатами по балансировке АФНФ [22–24] симметричные АФНФ не накладывают ограничения на количество и вид формируемых запросов и исключают процедуру обеспечения симметричности путей из процесса изготовления АФНФ.

1. Анализ АФНФ. При реализации АФНФ изготавливается множество функционально и топологически идентичных пар электрических путей, представляющих собой последовательно подключенные базовые элементы и их межсоединения. Для построения таких пар путей используются базовые элементы, которые реализуют три функции, а именно функцию генерирования (*Generate*) двух случайных величин добавленной временной задержки, функцию выбора (*Select*) одной из двух случайных величин и функцию переключения (*Switch*) путей [23, 24]. В классическом представлении j-й базовый элемент реализуется с использованием двух мультиплексоров MUX_{1j} и MUX_{2j} (см. рис. 1) [6, 8, 23, 24]. Соответственно, подобный базовый элемент выполняет три указанные ранее функции. Последовательно рассмотрим каждую из них.

Функция *Generate* реализует генерирование уникальных значений задержек как результат случайных факторов при производстве АФНФ. Каждый из двух мультиплексоров MUX_{1j}

и MUX_{2j} базового элемента представляет собой уникальный физический объект, описываемый характеристиками, имеющими фиксированные значения, но полученными как результат множества непредсказуемых и случайных факторов при его изготовлении. В первую очередь к таким характеристикам относятся величины задержек $\Delta(0)_{1,j}$, $\Delta(0)_{2,j}$, $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$. Численное значение $\Delta(0)_{1,j}$ определяет временную задержку прохождения сигнала с нулевого входа, обозначенного символом 0, для первого мультиплексора (MUX_{1j}) j-й ступени АФНФ на его выход, а $\Delta(0)_{2,j}$ – задержку для второго мультиплексора (MUX_{2j}). Величины $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$ представляют собой задержки сигналов по единичным входам соответствующих мультиплексоров. В процессе функционирования АФНФ эти величины в идеальном случае имеют отличающиеся, но неизменные значения и участвуют в определении величин добавленной разности задержек $\delta_{0,j}$ и $\delta_{1,j}$ согласно уравнениям

$$\delta_{0,j} = \Delta(0)_{1,j} - \Delta(0)_{2,j};\ \delta_{1,j} = \Delta(1)_{1,j} - \Delta(1)_{2,j}. \quad (1)$$

Разность задержки $\delta_{0,j}$ для j-й ступени АФНФ формируется при $c_j = 0$ как добавленная разность задержек $\Delta(0)_{1,j}$ и $\Delta(0)_{2,j}$ прохождения сигнала по двум путям через MUX_{1j} и MUX_{2j}, а разность задержки $\delta_{1,j}$ – как разность $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$ при $c_j = 1$. Значения величин добавленной разности задержек $\delta_{0,j}$ и $\delta_{1,j}$ являются результатом функции *Generate* j-го базового элемента.

Функция *Select* в соответствии со значением бита $c_j \in \{0, 1\}$ запроса $C_i = c_0\, c_1 \ldots c_{n-1}$ выбирает одну из двух величин добавленной разности задержек: $\delta_{0,j}$ или $\delta_{1,j}$. Аргументом этой функции является значение бита $c_j \in \{0, 1\}$ запроса C_i, который определяет одну из двух пар путей через j-й базовый элемент. Разница задержки d_j после j-ступени вычисляется в соответствии со следующим рекуррентным уравнением [8, 23]:

$$d_j = \delta_{c_j,j} + d_{j-1} \cdot (-1)^{c_j};\ d_{-1} = 0;\ j = 0,1,2,\ldots,n-1. \quad (2)$$

Как видно из соотношения (2), функция *Select* j-го базового элемента определяет первое слагаемое, которое принимает одно из двух значений: $\delta_{0,j}$ при $c_j = 0$ или $\delta_{1,j}$ при $c_j = 1$.

Аргументом функции *Switch*, как и функции *Select*, является значение c_j, которое определяет знак накопленной на предыдущих базовых элементах по отношению к j-му базовому элементу добавленной задержки d_{j-1} (2). При $c_j = 1$ выполняется переключение одного пути на j-м базовом элементе АФНФ с MUX_{1j} на MUX_{2j}, а второго – с MUX_{2j} на MUX_{1j}, что эквивалентно изменению знака разницы задержек сигналов пары путей на предыдущих ступенях АФНФ.

Суммарное значение разности задержек d_{n-1} по выбранной запросом C_i паре путей, а именно его знак плюс либо минус, и определяет ответ R_i на запрос C_i:

$$d_{n-1} = \delta_{c_{n-1},n-1} + \sum_{j=0}^{n-2}(\delta_{c_j,j} \cdot \prod_{k=j+1}^{n-1}(-1)^{c_k}). \quad (3)$$

Аналогично, как и для случая сбалансированных АФНФ [24], соотношение (3) для вычисления d_{n-1} может быть представлено в следующем виде:

$$d_{n-1} = \delta_{c_{n-1},n-1} + \sum_{j=0}^{n-2}(\delta_{c_j,j} \cdot \prod_{k=j+1}^{n-1}(1-2\cdot c_k)) = \delta_{c_{n-1},n-1} + \sum_{j=0}^{n-2}(\delta_{c_j,j} \cdot (1-2\cdot \bigoplus_{k=j+1}^{n-1} c_k)). \quad (4)$$

Выражение (4) представляет собой две формулы для вычисления величины d_{n-1}, отличные от (3), в которых используются арифметические операции +, – и ·, а также логическая операция сложения по модулю два $\oplus$. Значение разности задержки $\delta_{c_j,j}$ каждой ступени АФНФ в приведенных формулах входит со знаком плюс либо минус в зависимости от запроса C_i. Соотноше-

ния (3) и (4) отличаются друг от друга формулами вычисления знака величины $\delta_{c_j,j}$, $j=\overline{0,n-2}$. При этом отметим, что знак добавленной задержки $\delta_{c_{n-1},n-1}$ всегда положителен.

Например, значение d_{n-1} для $n = 4$ и $C_i = c_0\,c_1\,c_2\,c_3 = 1\ 0\ 0\ 1$ вычисляется с применением формулы (3) следующим образом:

$$d_3 = \delta_{c_0,0}\cdot(-1)^{c_1}\cdot(-1)^{c_2}\cdot(-1)^{c_3} + \delta_{c_1,1}\cdot(-1)^{c_2}\cdot(-1)^{c_3} + \delta_{c_2,2}\cdot(-1)^{c_3} + \delta_{c_3,3} = -\delta_{1,0} - \delta_{0,1} - \delta_{0,2} + \delta_{1,3}.$$

Аналогичный результат может быть получен на основании формулы (4):

$$d_3 = \delta_{c_0,0}\cdot(1-2(c_1\oplus c_2\oplus c_3)) + \delta_{c_1,1}\cdot(1-2(c_2\oplus c_3)) + \delta_{c_2,2}\cdot(1-2(c_3)) + \delta_{c_3,3} = -\delta_{1,0} - \delta_{0,1} - \delta_{0,2} + \delta_{1,3}.$$

Если представить знаки величин $\delta_{c_j,j}$ в выражениях (3) и (4) в виде вектора $B_i = b_0\,b_1\,b_2\,...\,b_{n-2}+1$, где $b_j\in\{+1, -1\}$, $j\in\{0, 1, 2, \ldots, n-2\}$, а $b_{n-1} = +1$, то соотношение, определяющее зависимость B_i от C_i, имеет следующий вид:

$$b_j = (1-2\cdot\overset{n-1}{\underset{k=j+1}{\oplus}} c_k),\ j=\overline{0,n-2};\ b_{n-1}=+1. \tag{5}$$

Принимая во внимание $b_j\in\{+1, -1\}$, можно заключить, что $(1-b_j)/2$ равняется 0 для $b_j = +1$ и 1 для $b_j = -1$. Отсюда следует, что для обеспечения знака +, т. е. значения $b_j = +1$, необходимо, чтобы для значений элементов запроса C_i выполнялось условие $\overset{n-1}{\underset{k=j+1}{\oplus}} c_k = 0$, а для обеспечения $b_j = -1$ это условие представляется как $\overset{n-1}{\underset{k=j+1}{\oplus}} c_k = 1$.

Таким образом, зависимость знаков +1 или –1 добавленных задержек $\delta_{c_j,j}$, $j=\overline{0,n-2}$, от запроса C_i определяется системой из $n - 1$ уравнений, в которой используются арифметические операции умножения и вычитания, а также логическая операция сложения по модулю два:

$$\begin{aligned}
&b_0 = 1-2\cdot(c_1\oplus c_2\oplus c_3\oplus\ldots\oplus c_{n-2}\oplus c_{n-1});\\
&b_1 = 1-2\cdot(c_2\oplus c_3\oplus c_4\oplus\ldots\oplus c_{n-2}\oplus c_{n-1});\\
&\ldots\\
&b_{n-3} = 1-2\cdot(c_{n-2}\oplus c_{n-1});\\
&b_{n-2} = 1-2\cdot c_{n-1}.
\end{aligned} \tag{6}$$

Исходными данными для системы (6) являются элементы $c_1\,c_2\,c_3\,...\,c_{n-1}$ вектора запроса C_i, а результатом – вектор знаков $B_i = b_0\,b_1\,b_2\,...\,b_{n-2}+1$. В дальнейшем символы +1 и –1 знаков b_j заменяются логическими значениями 0 и 1. Соответственно, $b_j = +1$ заменяется на логический ноль, а $b_j = -1$ – на логическую единицу. Например, вектор, состоящий из четырех знаков $B_i = b_0\,b_1\,b_2\,b_3 = +1\ +1\ -1\ -1$, представляется в виде двоичного вектора $B_i = b_0\,b_1\,b_2\,b_3 = 0\ 0\ 1\ 1$.

Значения векторов запроса $C_i = c_0\,c_1\,c_2\,c_3$ и соответствующих им векторов знаков $B_i = b_0\,b_1\,b_2\,b_3$ для $n = 4$ приведены в табл. 1. В столбцах таблицы дается описание трех АФНФ, каждая из которых построена на четырех базовых элементах согласно рис. 1. Первая из них, а именно $АФНФ_0$, представляет собой классическую АФНФ, для которой значения добавленных задержек $\delta_{0,j}$ и $\delta_{1,j}$ для каждого из четырех базовых элементов являются уникальными случайными величинами. Для $АФНФ_0$ и остальных АФНФ приведены значения добавленной разности задержки d_3 на выходах пары путей, знак которой определяет ответ R_i на запрос C_i (см. рис. 1) для $n = 4$. $АФНФ_1$ представляет случай, когда задержки $\Delta(0)_{1,j}$ и $\Delta(1)_{1,j}$ мультиплексора MUX_{1j}, а также задержки $\Delta(0)_{2,j}$ и $\Delta(1)_{2,j}$ мультиплексора MUX_{2j} одинаковы, т. е. $\Delta(0)_{1,j} = \Delta(1)_{1,j}$ и $\Delta(0)_{2,j} = \Delta(1)_{2,j}$. Тогда для добавленных задержек $\delta_{0,j}$ и $\delta_{1,j}$ согласно (1) выполняется равен-

ство $\delta_{0,j} = \delta_{1,j} = \delta_j$. Подобное соотношение задержек является весьма вероятным, учитывая симметрию мультиплексоров, для которых задержки по единичному и нулевому входам должны быть одинаковыми либо, в худшем случае, близкими по величине.

Пример АФНФ$_2$ рассматривался ранее в работе [23] как результат изготовления АФНФ, когда из-за вариаций производственного процесса не только равны задержки $\Delta(0)_{1,j}$ и $\Delta(1)_{1,j}$ мультиплексоров MUX_{1j} и задержки $\Delta(0)_{2,j}$ и $\Delta(1)_{2,j}$ мультиплексоров MUX_{2j} всех $n = 4$ ступеней АФНФ$_2$, но и их разность для всех базовых элементов принимает одинаковое значение d, т. е. $\delta_{0,j} = \delta_{1,j} = \delta_j = d$. Отмечалась реальность такой ситуации в технологических процессах изготовления подобных функций, особенно при реализации АФНФ на программируемых структурах [15, 16, 22].

Таблица 1
Описание функционирования АФНФ$_0$, АФНФ$_1$ и АФНФ$_2$

Table 1
Description of $APUF_0$, $APUF_1$ and $APUF_2$ functioning

$C_i = c_0\,c_1\,c_2\,c_3$	$B_i = b_0\,b_1\,b_2\,b_3$	АФНФ$_0$ $APUF_0$	АФНФ$_1$ $APUF_1$	АФНФ$_2$ $APUF_2$	R_i
		d_3	d_3	d_3	
0 0 0 0	0 0 0 0	$+\delta_{0,0}+\delta_{0,1}+\delta_{0,2}+\delta_{0,3}$	$+\delta_0+\delta_1+\delta_2+\delta_3$	$+d+d+d+d = 4d$	0
0 0 0 1	1 1 1 0	$-\delta_{0,0}-\delta_{0,1}-\delta_{0,2}+\delta_{1,3}$	$-\delta_0-\delta_1-\delta_2+\delta_3$	$-d-d-d+d = -2d$	1
0 0 1 0	1 1 0 0	$-\delta_{0,0}-\delta_{0,1}+\delta_{1,2}+\delta_{0,3}$	$-\delta_0-\delta_1+\delta_2+\delta_3$	$-d-d+d+d = 0$	X
0 0 1 1	0 0 1 0	$+\delta_{0,0}+\delta_{0,1}-\delta_{1,2}+\delta_{1,3}$	$+\delta_0+\delta_1-\delta_2+\delta_3$	$+d+d-d+d = 2d$	0
0 1 0 0	1 0 0 0	$-\delta_{0,0}+\delta_{1,1}+\delta_{0,2}+\delta_{0,3}$	$-\delta_0+\delta_1+\delta_2+\delta_3$	$-d+d+d+d = 2d$	0
0 1 0 1	0 1 1 0	$+\delta_{0,0}-\delta_{1,1}-\delta_{0,2}+\delta_{1,3}$	$+\delta_0-\delta_1-\delta_2+\delta_3$	$+d-d-d+d = 0$	X
0 1 1 0	0 1 0 0	$+\delta_{0,0}-\delta_{1,1}+\delta_{1,2}+\delta_{0,3}$	$+\delta_0-\delta_1+\delta_2+\delta_3$	$+d-d+d+d = 2d$	0
0 1 1 1	1 0 1 0	$-\delta_{0,0}+\delta_{1,1}-\delta_{1,2}+\delta_{1,3}$	$-\delta_0+\delta_1-\delta_2+\delta_3$	$-d+d-d+d = 0$	X
1 0 0 0	0 0 0 0	$+\delta_{1,0}+\delta_{0,1}+\delta_{0,2}+\delta_{0,3}$	$+\delta_0+\delta_1+\delta_2+\delta_3$	$+d+d+d+d = 4d$	0
1 0 0 1	1 1 1 0	$-\delta_{1,0}-\delta_{0,1}-\delta_{0,2}+\delta_{1,3}$	$-\delta_0-\delta_1-\delta_2+\delta_3$	$-d-d-d+d = -2d$	1
1 0 1 0	1 1 0 0	$-\delta_{1,0}-\delta_{0,1}+\delta_{1,2}+\delta_{0,3}$	$-\delta_0-\delta_1+\delta_2+\delta_3$	$-d-d+d+d = 0$	X
1 0 1 1	0 0 1 0	$+\delta_{1,0}+\delta_{0,1}-\delta_{1,2}+\delta_{1,3}$	$+\delta_0+\delta_1-\delta_2+\delta_3$	$+d+d-d+d = 2d$	0
1 1 0 0	1 0 0 0	$-\delta_{1,0}+\delta_{1,1}+\delta_{0,2}+\delta_{0,3}$	$-\delta_0+\delta_1+\delta_2+\delta_3$	$-d+d+d+d = 2d$	0
1 1 0 1	0 1 1 0	$+\delta_{1,0}-\delta_{1,1}-\delta_{0,2}+\delta_{1,3}$	$+\delta_0-\delta_1-\delta_2+\delta_3$	$+d-d-d+d = 0$	X
1 1 1 0	0 1 0 0	$+\delta_{1,0}-\delta_{1,1}+\delta_{1,2}+\delta_{0,3}$	$+\delta_0-\delta_1+\delta_2+\delta_3$	$+d-d+d+d = 2d$	0
1 1 1 1	1 0 1 0	$-\delta_{1,0}+\delta_{1,1}-\delta_{1,2}+\delta_{1,3}$	$-\delta_0+\delta_1-\delta_2+\delta_3$	$-d+d-d+d = 0$	X

Ответы R_i на каждый из запросов C_i для АФНФ, приведенных в табл. 1, определяются знаком добавленной задержки d_3. Предположив, что для АФНФ$_2$ задержка d принимает положительное значение, получим ответ $R_i = 0$ для положительных d_3, $R_i = 1$ – для отрицательных d_3 и метастабильный ответ $R_i = X$ для $d_3 = 0$. Анализ приведенных в табл. 1 данных свидетельствует об асимметрии множества ответов R_i, что негативно сказывается на вероятностных характеристиках АФНФ. В случае АФНФ$_2$ вероятность формирования ответа $R_i = 0$ оказывается существенно больше вероятности появления ответа $R_i = 1$.

Приведенные аналитические соотношения, описывающие функционирование АФНФ, а также примеры АФНФ для $n = 4$ позволяют сформулировать следующее утверждение.

Утверждение 1. *Математическое ожидание* $\mu(d_{n-1})$ *добавленной задержки* d_{n-1} *классической АФНФ равняется* $(\delta_{0,n-1} + \delta_{1,n-1})/2$.

Доказательство. Запрос C_i для АФНФ, представляющей собой n последовательно подключенных базовых элементов, формируется случайным образом с вероятностью $p(C_i) = 1/2^n$. Каждому запросу C_i соответствует ответ R_i в виде добавленной задержки $d_{n-1}(C_i)$ (3). Тогда $\mu(d_{n-1})$ определяется согласно соотношению

$$\mu(d_{n-1}) = \sum_{i=0}^{2^n-1} d_{n\text{-}1}(C_i)\cdot p(C_i) = \frac{1}{2^n}\sum_{i=0}^{2^n-1} d_{n\text{-}1}(C_i). \qquad (7)$$

Величина $d_{n-1}(C_i)$ определяется алгебраической суммой добавленных задержек $b_j \cdot \delta_{c_j,j}$, $j=\overline{0,n-1}$ базовых элементов АФНФ согласно (3). Значение бита c_j, $j=\overline{0,n-2}$, запроса C_i определяет выбор одной из двух задержек $\delta_{0,j}$ или $\delta_{1,j}$, а биты $c_{j+1}\ c_{j+2}\ c_{j+3}\ \ldots\ c_{n-1}$ формируют ее знак b_j, равный +1 или –1 (5). Значения бит $c_j\ c_{j+1}\ c_{j+2}\ \ldots\ c_{n-1}$ запроса C_i принимают все возможные из 2^{n-j} комбинации для каждого двоичного кода $c_0\ c_1\ c_2\ \ldots\ c_{j-1}$ в младших его разрядах. Соответственно, ровно половина задержек $\delta_{c_j,j}$, входящих в выражение $d_{n-1}(C_i)$, примет значение $\delta_{0,j}$, а вторая – $\delta_{1,j}$. Это следует из того факта, что половина значений c_j для всех возможных комбинаций $c_0\ c_1\ c_2\ \ldots\ c_{j-1}$ равняется нулю, а вторая половина – единице. В силу того что в выражении (7) двоичное представление кода $c_{j+1}\ c_{j+2}\ \ldots\ c_{n-1}$ также принимает все возможные комбинации, то, соответственно, согласно (5) половина значений $\delta_{0,j}$ будет иметь знак +1 и столько же знак –1. Аналогично это справедливо и для $\delta_{1,j}$. Исключение составляют задержки $\delta_{0,n-1}$ и $\delta_{1,n-1}$ последнего базового элемента, которые имеют только знак +1. Таким образом, в выражении (7) суммируются все возможные значения $d_{n-1}(C_i)$, каждое из которых состоит из n слагаемых $b_j \cdot \delta_{c_j,j}$, $j=\overline{0,n-1}$. При этом половина $\delta_{0,j}$, $j=\overline{0,n-2}$, и половина $\delta_{1,j}$, $j=\overline{0,n-2}$, будут иметь знак $b_j = +1$, а вторые половины – знак $b_j = -1$, и только $\delta_{0,n-1}$ и $\delta_{1,n-1}$ входят в эту сумму со знаком +1. Все множество слагаемых $d_{n-1}(C_i)$ и их составляющих $b_j \cdot \delta_{c_j,j}$, $j=\overline{0,n-1}$, для $n = 4$ приведено в табл. 1 (см. $АФНФ_0$). Окончательно выражение для $\mu(d_{n-1})$ равняется $(\delta_{0,n-1} + \delta_{1,n-1})/2$, что и требовалось доказать.

Значение $\mu(d_{n-1})$ для $АФНФ_0$ принимает вид $(\delta_{0,3} + \delta_{1,3})/2$, что соответствует утверждению 1, для $АФНФ_1$ оно составляет δ_3, а для $АФНФ_2$ его величина равняется d (см. табл. 1). В идеальном случае для обеспечения равной вероятности получения для АФНФ нулевого и единичного ответов R_i необходимо, чтобы $\mu(d_{n-1}) = 0$.

Приведенный анализ АФНФ позволяет сделать следующие выводы. В первую очередь необходимо отметить удачный набор трех функций *Generate*, *Select* и *Switch*, реализуемых базовым элементом классической АФНФ [23]. Их сочетание обеспечивает формирование случайных значений добавленных задержек, выбор одной из них со знаком плюс либо минус осуществляется в соответствии с запросом C_i. Результирующая сумма (3) таких задержек на n последовательно соединенных базовых элементах определяет ответ R_i, который имеет достаточно хорошие характеристики, отмеченные в большинстве публикаций [8, 15, 23]. Именно сочетание указанных функций обеспечило вполне работоспособное состояние $АФНФ_2$, несмотря на исключительно аномальное сочетание добавленных задержек, которые зачастую не имеют ничего общего со случайными значениями, как это, например, показано для случая $АФНФ_2$.

Необходимо отметить и недостатки классической АФНФ, которые в первую очередь связаны с асимметрией пар путей, вызванной в том числе ненулевым значением $\mu(d_{n-1})$ (см. утверждение 1). Эффект асимметрии сказывается на том, что поведение классических АФНФ отличается от желаемого, особенно при их реализации на программируемой логике типа FPGA, что объясняется сложностью, а в большинстве случаев и невозможностью обеспечения физической идентичности элементов и симметричности их межсоединений [22]. Зачастую наблюдается абсолютная асимметрия, требующая дальнейшей балансировки путей, что относится к нежелательной, но вынужденной процедуре [22, 24]. Данная процедура для АФНФ, приведенных в табл. 1, будет состоять в нивелировании влияния добавленной задержки третьего базового элемента путем добавления по одному из его входов линии задержки. Величина значения временной задержки, используемой для балансировки $АФНФ_0$, равняется $(\delta_{0,3} + \delta_{1,3})/2$. Для $АФНФ_1$ она составляет δ_3, для $АФНФ_2$ – d. Во всех трех случаях после балансировки будет выполняться равенство $\mu(d_3) = 0$, что обеспечит симметричность выбираемых пар путей и, соответственно, улучшение основных характеристик АФНФ.

Эффект симметричности проявляется на уровне конкретных скорректированных ответов в виде добавленной задержки $d_3(C_i)$. Например, на запрос $C_i = 0\ 0\ 1\ 1$ сбалансированная $АФНФ_0$ генерирует задержку $d_3(0011) = +\delta_{0,0} +\delta_{0,1}–\delta_{1,2} +\delta_{1,3} – (\delta_{0,3} + \delta_{1,3})/2 = +\delta_{0,0} +\delta_{0,1}–\delta_{1,2}– \delta_{0,3}/2 + \delta_{1,3}/2$. В соответствии с утверждением 1 для запроса $C_i = 0\ 0\ 1\ 1$ существует запрос C_k, $i \neq k$,

для которого формируется значение $d_3(C_k)$, равное по абсолютной величине $d_3(0011)$, но имеющее противоположный знак. Действительно, для C_k = 0 0 1 0 получаем $d_3(0010) = -\delta_{0,0} -\delta_{0,1}+\delta_{1,2} +\delta_{0,3} - (\delta_{0,3} + \delta_{1,3})/2 = -\delta_{0,0} -\delta_{0,1}+\delta_{1,2} +\delta_{0,3}/2- \delta_{1,3}/2$. Видно, что $d_3(0011)$ и $d_3(0010)$ равны по абсолютной величине $|d_3(0011)| = |d_3(0010)|$, но имеют противоположные знаки. Это свидетельствует о симметричности относительно нулевого среднего $\mu(d_3) = 0$. Очевидно, что для общего случая сбалансированных АФНФ, в том числе и для всех трех сбалансированных АФНФ, приведенных в табл. 1, для запроса C_i всегда существует запрос C_k. Взаимосвязь C_i и C_k определяется равенством

$$d_{n-1}(C_i) = -d_{n-1}(C_k);\ i \neq k;\ i,k \in \{0,1,2,\ldots n-1\}. \quad (8)$$

Балансировка является одним из наиболее эффективных методов увеличения стабильности АФНФ [17, 22, 24]. Однако эта процедура, требующая дополнительных индивидуальных настроек АФНФ, технологически может быть сложной задачей, а в ряде случаев ASIC-технологий и невыполнимой. Балансировка путей означает отход от основополагающей концепции ФНФ, заключающейся в использовании при изготовлении ФНФ их единого схемотехнического описания для получения неповторяемого поведения ФНФ, описываемого уникальной функцией $R = F(C)$ [6, 8].

2. Математическая модель симметричных АФНФ. Анализ, проведенный в предыдущем разделе, показал, что в сбалансированной АФНФ наблюдается симметрия формируемых добавленных задержек d_{n-1} относительно их нулевого математического ожидания. Знак величины d_{n-1} плюс либо минус определяет ответ $R_i \in \{0, 1\}$, для которого и необходимо выполнение условия равной вероятности двух возможных ответов.

Приведенная формулировка поведения симметричных АФНФ напоминает классическую модель, описывающую подбрасывание монеты. Можно предположить, что структура АФНФ представляет собой монету, а подаваемые на нее запросы C_i имитируют ее подбрасывание, в результате которого формируются ответы $R_i = 0$ (орел) и $R_i = 1$ (решка). В классической постановке задачи результатом подбрасывания монеты являются независимые величины ответов (орел или решка), что нельзя утверждать в случае АФНФ. Более того, важным свойством АФНФ является повторяемость результатов эксперимента, заключающаяся в том, что для повторяемого запроса C_i ответ R_i должен быть таким же, т. е. повторяемым. В то же время уникальность, характеризующаяся различными значениями ответов R_i на одни и те же запросы C_i для физически различных АФНФ, определяется их структурой и внутренними параметрами (монетой). Что касается параметров структуры АФНФ, то их задание определяется множеством случайных факторов при изготовлении АФНФ, однако при проведении экспериментов эти параметры в идеальном случае должны быть неизменными.

Принимая во внимание приведенные замечания, а также анализ классической АФНФ, изложенный в разд. 1, можно заключить, что математическая модель функционирования симметричных АФНФ описывается моделью псевдоподбрасывания монеты. Единственным отличием модели псевдоподбрасывания монеты от классической модели подбрасывания является повторяемость результатов подбрасывания для идентичных подбрасываний, определяемых одним и тем же запросом C_i. Все остальные свойства псевдоподбрасывания должны быть такими же либо максимально близкими к свойствам процедуры подбрасывания монеты. Соотношение двух указанных математических моделей аналогично соотношению моделей генерирования псевдослучайных и случайных чисел [25].

Формулировка обобщенной математической модели, описывающей симметричные АФНФ, как аналога классической модели псевдоподбрасывания монеты принимает следующий вид:

1. Структура АФНФ (монета) однозначно задается множеством $\delta_0, \delta_1, \delta_2, \ldots, \delta_{n-1}$ произвольных, сгенерированных случайным образом величин добавленной разности задержек. Отметим, что значения $\delta_0, \delta_1, \delta_2, \ldots, \delta_{n-1}$ не изменяются в процессе функционирования АФНФ.

2. Запросы C_i, имитирующие подбрасывание монеты, формируются случайным образом в виде двоичного вектора $C_i = c_0\, c_1\, c_2 \ldots c_{n-1}$, где $c_j \in \{0, 1\}$, j = 0, 1, 2, …, n–1, а $p(c_j = 0) = p(c_j = 1) = 0{,}5$.

3. Процедура имитации подбрасывания монеты заключается в вычислении суммы d_{n-1}:

$$d_{n-1} = (1-2\cdot c_0)\cdot\delta_0 + (1-2\cdot c_1)\cdot\delta_1 + (1-2\cdot c_2)\cdot\delta_2 + \ldots + (1-2\cdot c_{n-1})\cdot\delta_{n-1}. \quad (9)$$

Выражение $(1-2c_j)$ представляет собой знак $b_j \in \{+1, -1\}$ слагаемого δ_j вектора знаков $B_i = b_0 b_1 b_2 \ldots b_{n-1}$ слагаемых суммы (9).

4. Значение знака суммы d_{n-1}, соответственно +1 или −1, определяет ответ $R_i = 0$ или $R_i = 1$ (орел или решка). Так как значения $\delta_0, \delta_1, \delta_2, \ldots, \delta_{n-1}$ не изменяются в процессе функционирования АФНФ, повторение запроса C_i для $d_{n-1} \neq 0$ приведет к повторению значения ответа R_i. В случае равенства нулю суммы d_{n-1} результатом будет метастабильное значение X ответа R_i. Оно характеризуется получением как нулевого, так и единичного результата при повторении одного и того же запроса C_i.

Представленная математическая модель характеризуется свойством симметричности получаемых значений суммы d_{n-1}, что является следствием следующего утверждения.

Утверждение 2. *Для математического ожидания* $\mu(d_{n-1})$ *суммы* d_{n-1}, *полученной согласно (9), справедливо равенство* $\mu(d_{n-1}) = 0$.

Доказательство. Запросы C_i для получения суммы (9) согласно описанной ранее обобщенной математической модели формируются случайным образом с вероятностью $p(C_i) = 1/2^n$. Согласно этой модели для запроса C_i формируется ответ $R_i(C_i)$ в виде знака +1 или −1 суммы d_{n-1}. В то же время для каждого запроса C_i существует инверсный запрос C_k, компоненты которого $\overline{c_0}\,\overline{c_1}\,\overline{c_2}\ldots\overline{c_{n-1}}$ принимают инверсные значения по отношению к компонентам $c_0 c_1 c_2 \ldots c_{n-1}$ запроса C_i. Использование запроса C_k приведет к инвертированию знаков слагаемых в выражении (9) и, соответственно, знака d_{n-1}. Таким образом, $d_{n-1}(C_i) = -d_{n-1}(C_k)$, что свидетельствует о выполнении условия симметричности (8), для которого сумма (7), определяющая значение математического ожидания $\mu(d_{n-1})$, равняется нулю. Что и требовалось доказать.

АФНФ, для которых выполняется свойство симметрии (8), обозначим как С_АФНФ. Примеры подобных функций для случая $n = 4$ и различных значений $\delta_0, \delta_1, \delta_2, \delta_3$ приведены в табл. 2.

Таблица 2
Описание функционирования симметричных С_АФНФ$_0$, С_АФНФ$_1$ и С_АФНФ$_2$

Table 2
Description of symmetric C_APUF$_0$, C_APUF$_1$ and C_APUF$_2$ functioning

$C_i = c_0\,c_1\,c_2\,c_3$	С_АФНФ$_0$ *C_APUF$_0$*		С_АФНФ$_1$ *C_APUF$_1$*		АФНФ$_1$ *APUF$_1$*		С_АФНФ$_2$ *C_APUF$_2$*	
	$\delta_0, \delta_1, \delta_2, \delta_3 =$ $= +1, -2, +3, -4$		$\delta_0, \delta_1, \delta_2, \delta_3 =$ $= +1, +2, +3, +4$		$\delta_0, \delta_1, \delta_2, \delta_3 =$ $= +1, +2, +3, +4$		$\delta_0, \delta_1, \delta_2, \delta_3 =$ $= +2, +2, +2, +2$	
	d_3	R_i	d_3	R_i	d_3	R_i	d_3	R_i
0 0 0 0	+1−2+3−4 =−2	1	+1+2+3+4 =+10	0	+1+2+3+4 =+10	0	+2+2+2+2 =+8	0
0 0 0 1	+1−2+3+4 =+6	0	+1+2+3−4 =+2	0	−1−2−3+4 =−2	1	+2+2+2−2 =+4	0
0 0 1 0	+1−2−3−4 =−8	1	+1+2−3+4 =+4	0	−1−2+3+4 =+6	0	+2+2−2+2 =+4	0
0 0 1 1	+1−2−3+4 = 0	X	+1+2−3−4 =−4	1	+1+2−3+4 =+4	0	+2+2−2−2 = 0	X
0 1 0 0	+1+2+3−4 =+2	0	+1−2+3+4 =+6	0	−1+2+3+4 =+8	0	+2−2+2+2 =+4	0
0 1 0 1	+1+2+3+4 =+10	0	+1−2+3−4 =−2	1	+1−2−3+4 = 0	X	+2−2+2−2 = 0	X
0 1 1 0	+1+2−3−4 =−4	1	+1−2−3+4 = 0	X	+1−2+3+4 = +6	0	+2−2−2+2 = 0	X
0 1 1 1	+1+2−3+4 =+4	0	+1−2−3−4 =−8	1	−1+2−3+4 =+2	0	+2−2−2−2 =−2	1
1 0 0 0	−1−2+3−4 =−4	1	−1+2+3+4 =+8	0	+1+2+3+4 =+10	0	−2+2+2+2 =+4	0
1 0 0 1	−1−2+3+4 =+4	0	−1+2+3−4 = 0	X	−1−2−3+4 = −2	1	−2+2+2−2 = 0	X
1 0 1 0	−1−2−3−4 =−10	1	−1+2−3+4 =+2	0	−1−2+3+4 =+4	0	−2+2−2+2 = 0	X
1 0 1 1	−1−2−3+4 =−2	1	−1+2−3−4 =−6	1	+1+2−3+4 =+4	0	−2+2−2−2 =−4	1
1 1 0 0	−1+2+3−4 = 0	X	−1−2+3+4 =+4	0	−1+2+3+4 =+8	0	−2−2+2+2 = 0	X
1 1 0 1	−1+2+3+4 =+8	0	−1−2+3−4 =−4	1	+1−2−3+4 = 0	X	−2−2+2−2 =−2	1
1 1 1 0	−1+2−3−4 =−6	1	−1−2−3+4 =−2	1	+1−2+3+4 = 6	0	−2−2−2+2 =−4	1
1 1 1 1	−1+2−3+4 =+2	0	−1−2−3−4 =−10	1	−1+2−3+4 =+2	0	−2−2−2−2 =−10	1

В табл. 2 также представлена реализация классической АФНФ$_1$ с такими же значениями δ_0, δ_1, δ_2, δ_3, = +1, +2, +3, +4, как и для С_АФНФ$_1$. Сравнение результатов поведения С_АФНФ$_1$ и АФНФ$_1$ показывает преимущества новой обобщенной модели симметричных АФНФ, обеспечивающих абсолютную симметрию ответов.

Приведенные конкретные реализации модели симметричных АФНФ позволяют сделать вывод об эффективности предложенной модели подобных АФНФ. Как видно из табл. 1, С_АФНФ$_0$, С_АФНФ$_1$ и С_АФНФ$_2$ характеризуются идеальной симметрией по сравнению с классической АФНФ$_1$.

3. Практическая реализация симметричных АФНФ. Множество разновидностей АФНФ представлено различными их модификациями для реализации – как схемой ASIC, так и схемой с использованием современных FPGA. В публикациях [8, 15, 17, 22, 24] указано, что идентичность элементов АФНФ, симметричность их геометрических параметров и межсоединений являлись и являются основополагающим требованием к подобным структурам. В то же время, понимая сложность, а в большинстве случаев и невозможность достижения топологической симметричности, в особенности межсоединений для FPGA, создаются и широко применяются АФНФ на программируемой логике. По мнению авторов, это объясняется тем, что значения величин добавленной разности задержек δ_0, δ_1, δ_2, …, δ_{n-1} в принципе могут быть произвольными, в том числе и детерминированными. Для любых значений этих величин, как следует из примеров, приведенных в табл. 1 и 2, достигается необходимая уникальность АФНФ. Однако закономерные зависимости между разностями задержек δ_0, δ_1, δ_2, …, δ_{n-1} приводят к уязвимости для взлома АФНФ путем применения методов машинного обучения [20, 21].

Требование формирования множества произвольных, сгенерированных случайным образом величин добавленной разности задержек δ_0, δ_1, δ_2, …, δ_{n-1} остается центральным для АФНФ. Попытки уменьшить зависимость между значениями задержек представлены в ряде работ [22, 24, 26, 27], которые направлены на совершенствование реализации функции *Generate* базового элемента АФНФ. Применение линий задержки с величинами задержки, существенно превышающими временные задержки распространения сигналов через элементы АФНФ и ее межсоединения, позволяет нивелировать топологическую асимметричность АФНФ [24]. Это, например, достигается включением по входам классического базового элемента АФНФ r последовательно подключенных логических элементов с целью увеличения значений задержек по каждому из двух входов базового элемента [24].

Реализация базового элемента АФНФ, использующего линии задержки *Del*1 и *Del*2, приведена на рис. 2, *а* [24]. Задержки $\Delta_{1,l}(j)$ и $\Delta_{2,l}(j)$, где $l \in \{0, 1, \ldots, r-1\}$, на каждом из r элементов линий задержки *Del*1 и *Del*2 по входам j-го базового элемента представляют собой значение случайной и независимой величины. Отметим, что для элементов, образующих линии задержки *Del*1 и *Del*2, например повторителей, в процессе их производства величина этих задержек случайным образом принимает конкретное значение. Этот факт и определяет уникальность и неповторяемость АФНФ. Задержка $\Delta_{v,l}(j)$, $v \in \{1, 2\}$, описывается математическим ожиданием (средним значением) $\mu(\Delta_{v,l}(j))$ и дисперсией (отклонением значений) $Var(\Delta_{v,l}(j))$. Величины добавленной разности δ_j задержек $\Omega1(j)$ и $\Omega2(j)$ последовательно соединенных логических элементов (например, повторителей) j-го базового элемента определяются как $\delta_j = \Omega1(j) - \Omega2(j) = [\Delta_{1,0}(j) + \Delta_{1,1}(j) + \ldots + \Delta_{1,r-1}(j)] - [\Delta_{2,0}(j) + \Delta_{2,1}(j) + \ldots + \Delta_{2,r-1}(j)]$. Мерой разброса значений добавленной разности задержек δ_j, которая является линейной комбинацией случайных величин $\Delta_{v,l}(j)$, будет значение дисперсии $Var(\delta_j) = 2r \times Var(\Delta_{v,l}(j))$. Среднеквадратическое (стандартное) отклонение $\sigma = \sqrt{2r \times Var(\Delta_{v,l}(j))}$ разности задержек δ_j растет с ростом величины r, которая определяет диапазон 3σ ее изменения относительно математического ожидания.

В рассмотренной структуре базового элемента (рис. 2, *а*) мультиплексоры выполняют только одну функцию, а именно *Switch*, так как величины задержек $\Omega1(j)$ и $\Omega2(j)$ на схемах *Del*1 и *Del*2 являются доминирующими, определяют значение δ_j и, соответственно, реализуют

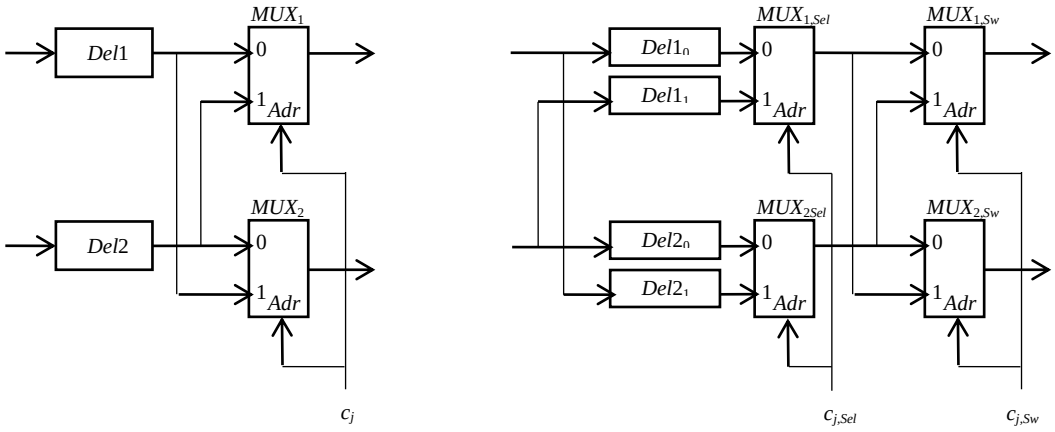
функцию *Generate*. Как показано в работе [24], описание функционирования *j*-го базового элемента АФНФ на линиях задержки отличается от описания (2) классической АФНФ и принимает вид

$$d_j = (d_{j-1} + \delta_j) \times (-1)^{c_j}. \tag{10}$$

Суммарное значение разности задержек d_{n-1} (11) по выбранной запросом C_i паре путей, аналогичное выражению (4), определяет ответ R_i на запрос C_i:

$$d_{n-1} = \sum_{j=0}^{n-1} \delta_j \times \prod_{i=j}^{n-1} (-1)^{c_i} = \sum_{j=0}^{n-1} \delta_j \times \prod_{i=j}^{n-1} (1 - 2 \times c_i) = \sum_{j=0}^{n-1} \delta_j \times (1 - 2 \times \bigoplus_{i=j}^{n-1} c_i). \tag{11}$$

Выражение (11) для вычисления d_{n-1} полностью соответствует реализации модели симметричных АФНФ, имитирующей псевдоподбрасывание монеты (9). В обоих случаях знак слагаемых, представляющих собой добавленные задержки δ_j, равновероятно принимает значения + и −.



a) *b)*

Рис. 2. Функциональные схемы базового элемента: *a)* сбалансированной АФНФ; *b)* симметричной АФНФ

Fig. 2. Functional diagrams of the base element: a) balanced PUF; b) symmetric PUF

Использование схем *Del*1 и *Del*2 в классическом базовом элементе позволяет нивелировать влияние задержек на межсоединениях и мультиплексорах [24]. Это достигается тем, что величины задержки сигнала $\Omega1(j)$ и $\Omega2(j)$ на схемах *Del*1 и *Del*2 задаются существенно бóльшими по сравнению с задержками на мультиплексорах $\Delta(0)_{1,j}$, $\Delta(0)_{2,j}$, $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$ и межсоединениях между ними. Количество *r* элементов в линиях задержки *Del*1 и *Del*2 может быть произвольным и по необходимости увеличиваться для обеспечения требуемых соотношений задержек и отклонений тестового сигнала.

В структуре базового элемента сбалансированной АФНФ (рис. 2, *a*) [24] мультиплексоры по сравнению с классическим базовым элементом выполняют только одну функцию, а именно *Switch*, и не выполняют функцию *Select*. Соответственно, функция *Generate* выполняется на схемах *Del*1 и *Del*2. Ее результатом является одна случайная величина δ_j, определяемая разностью $\Omega1(j) - \Omega2(j)$ и имеющая знак + при $c_j = 0$ и знак − при $c_j = 1$. Высокая степень сбалансированности рассмотренных в работе [24] АФНФ позволяет достичь приемлемого уровня стабильности, единообразия и внутрикристальной уникальности АФНФ. Это в первую очередь достигается обеспечением симметричности гарантированной математической моделью псевдоподбрасывания монеты, описываемой соотношением (9). Однако отсутствие операции *Select* влияет на уникальность АФНФ, определяемую случайными факторами изготовления АФНФ.

Отметим, что в классической реализации базового элемента все три функции, а именно *Generate*, *Select* и *Switch*, выполняются на двух мультиплексорах, что приводит к асимметрии АФНФ (см. утверждение 1).

В качестве базового элемента симметричных АФНФ (С_АФНФ) используем структуру, объединяющую достоинства как классического базового элемента, так и базового элемента сбалансированных АФНФ. Функционально такой элемент выполняет все три функции классического элемента и обеспечивает условие симметричности пар путей, описываемое утверждением 2. Реализация базового элемента симметричных АФНФ показана на рис. 2, *b*.

Предлагаемый базовый элемент симметричных АФНФ выполняет все три функции классического базового элемента. В то же время в отличие от классических и сбалансированных АФНФ все эти функции выполняются на различных компонентах базового элемента.

Функция *Generate* реализуется на линиях задержки $Del1_0$, $Del1_1$, $Del2_0$ и $Del2_1$ с задержками $\Omega1_0(j)$, $\Omega1_1(j)$, $\Omega2_0(j)$ и $\Omega2_1(j)$, которые должны быть несравнимо больше задержек на мультиплексорах. Таким образом, временные задержки мультиплексоров практически не влияют на величины генерируемых добавленных разностей задержки. Аналогично, как и для классической АФНФ (1), эти величины определяются из соотношений

$$\delta_{0,j} = \Omega1_0(j) - \Omega2_0(j); \quad \delta_{1,j} = \Omega1_1(j) - \Omega2_1(j). \tag{12}$$

Функцию *Select* выполняют два мультиплексора $MUX_{1,Sel}$ и $MUX_{2,Sel}$, которые в зависимости от значения бита запроса $c_{j,Sel} \in \{0, 1\}$ выбирают одну из двух случайных величин добавленных разностей задержек $\delta_{0,j}$ или $\delta_{1,j}$. В рамках рассмотренной ранее математической модели симметричных АФНФ данная функция определяет структуру монеты (см. разд. 2), которая описывается множеством задержек $\delta_{c_{0,Sel},0}, \delta_{c_{1,Sel},1}, \delta_{c_{2,Sel},2}, ..., \delta_{c_{m-1,Sel},n-1}$. Таким образом, биты запроса $c_{0,Sel}\, c_{1,Sel}\, c_{2,Sel}$ и $c_{n-1,Sel}$, по сути, выбирают одну их 2^n возможных монет, для которых и реализуется процедура псевдоподбрасывания.

Два следующих мультиплексора $MUX_{1,Sw}$ и $MUX_{2,Sw}$ каждого базового элемента отвечают за реализацию функции *Switch*, выполняющей переключение пары путей (она же монета), выбранной битами запроса $c_{j,Sel}$. В терминах математической модели псевдоподбрасывания монеты функция *Switch* реализует подбрасывание монеты согласно (9), т. е. определяет знаки слагаемых $\delta_{c_{0,Sel},0}, \delta_{c_{1,Sel},1}, \delta_{c_{2,Sel},2}, ..., \delta_{c_{m-1,Sel},n-1}$, выбранных функцией *Select*. Аргументом этой функции j-го базового элемента (рис. 1, *b*) является бит запроса $c_{j,Sw} \in \{0, 1\}$. Так же, как и в случае классического базового элемента, на значение знаков слагаемых, представляющих собой добавленные разности задержки, оказывают влияние и биты запроса $c_{j,Sel}$ на знаки указанных величин, но только для предыдущих базовых элементов по отношению к текущему j-му. Бит запроса $c_{j,Sw}$ влияет на знак добавленной разности задержки j-го базового элемента непосредственно, как это видно из рекуррентного выражения

$$d_j = (d_{j-1} \cdot (-1)^{c_{j,Sel}} + \delta_{c_{j,Sel},j}) \cdot (-1)^{c_{j,Sw}} = d_{j-1} \cdot (-1)^{c_{j,Sel} \oplus c_{j,Sw}} + \delta_{c_{j,Sel},j} \cdot (-1)^{c_{j,Sw}};$$
$$d_{-1} = 0; \; j = 0,1, \ldots, n-1.$$

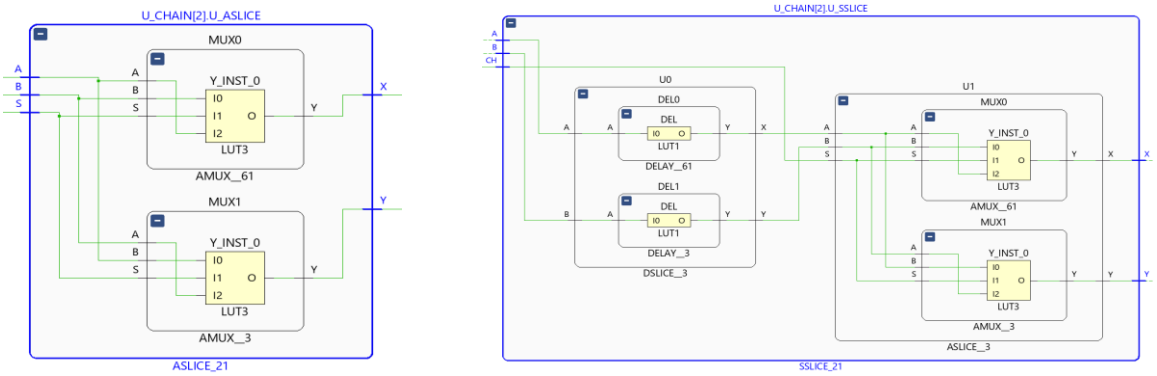
Приведенное выражение объединяет соотношения (2) и (10) в связи с тем, что предложенный новый базовый элемент включает основные свойства как классической АФНФ, так и сбалансированной. В части классической АФНФ новый базовый элемент реализует все три его функции, в том числе весьма значимую функцию *Switch*. В то же время идеи, заложенные в сбалансированных АФНФ, нашли свое отражение в обеспечении симметрии пар путей, выбираемых запросами в С_АФНФ.

Суммарное значение разности задержек d_{n-1} С_АФНФ по выбранной запросом C_i паре путей, а именно его знак + либо –, и определяет ответ R_i на запрос C_i:

$$d_{n-1} = \delta_{c_{(n-1),Sel},n-1} \cdot (-1)^{c_{(n-1),Sw}} + \sum_{j=0}^{n-2} (\delta_{c_{j,Sel},j} \cdot \prod_{k=j+1}^{n-1} (-1)^{c_{k,Sel} \oplus c_{k,Sw}}). \tag{13}$$

Выражение (13) для вычисления d_{n-1}, так же как и соотношение (11), соответствует реализации модели симметричных АФНФ, имитирующей псевдоподбрасывание монеты (9), так как знак слагаемых, представляющих собой выбранные функцией *Select* добавленные задержки $\delta_{c_{j,Sel},j}$, равновероятно принимает значение + и −.

4. Описание экспериментальных исследований. Для подтверждения эффективности предложенных в статье новых решений по построению симметричных АФНФ был проведен ряд экспериментов на программируемых логических интегральных схемах FPGA Xilinx Zynq7, входящих в состав плат быстрого прототипирования цифровых устройств Digilent Zybo Z7-10. Реализовывались четыре идентичных экземпляра классической схемы АФНФ и четыре экземпляра симметричной схемы АФНФ с применением базового элемента, представленного на рис. 2, *a*, для $n = 32$. На рис. 3 приведены примеры реализации базового элемента как для классической схемы АФНФ, так и для симметричной схемы в терминах технологических блоков FPGA. Элементы задержки для симметричной схемы АФНФ были реализованы на LUT-блоках в качестве логических повторителей сигналов (элементы LUT1 на рис. 3, *b*).



a) *b)*

Рис. 3. Базовый элемент для классической схемы АФНФ (*a*) и симметричной схемы АФНФ (*b*)

Fig. 3. Basic element for classical APUF scheme (a) and symmetrical APUF scheme (b)

В ходе экспериментов исследовались временные параметры множества пар путей, а именно значения d_{31} для различных запросов, выраженных как $\Delta_{C_i}^f$, где $f \in [0,3]$ есть индекс экземпляра одной из четырех исследуемых схем АФНФ, реализованных на одном кристалле FPGA. Отсортированные результаты указанных параметров приведены на рис. 4.

В отличие от проведенных ранее экспериментов [24] значения подаваемых запросов не подвергались балансировке, а полученные данные по основным характеристикам АФНФ коррелируются с приведенными ниже.

Значения математического ожидания $\mu(\Delta_{C_i}^f)$ (см. табл. 3) для 10^4 различных запросов, полученных с помощью 32-разрядого генератора М-последовательностей, показывают асимметрию во всех четырех реализациях классической АФНФ, в то время как для реализаций симметричной АФНФ свидетельствуют об их большей симметричности.

Таблица 3

Математическое ожидание $\mu(\Delta_{C_i}^f)$, нс

Table 3

Expected value $\mu(\Delta_{C_i}^f)$, *ns*

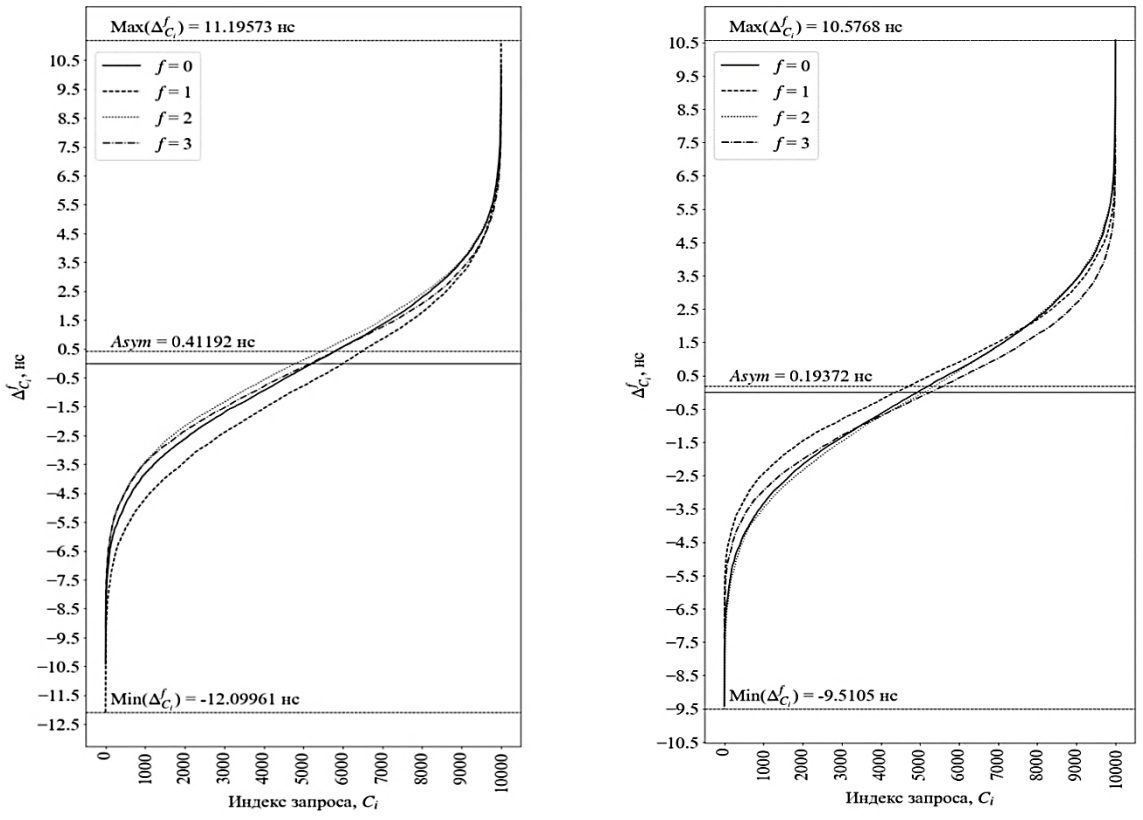
f	Классическая АФНФ *Classic APUF*	Симметричная АФНФ *Symmetrical APUF*
0	–0,1739	0,0212
1	–0,7899	0,3562
2	0,0927	–0,0455
3	–0,1259	–0,1437

Для более детального сравнения реализованных схем АФНФ были определены такие их характеристики, как единообразие (*Un*) и внутрикристальная уникальность (U_{intra}) [16, 22]. Значения единообразия и внутрикристальной уникальности для симметричной АФНФ превышают аналогичные значения для классической АФНФ (табл. 4).

Таблица 4
Усредненные значения *Un* и U_{intra}
Table 4
Average Un and U_{intra}

Тип базового элемента *Base element type*	*Un*	U_{intra}
Классическая схема АФНФ	0,9185	0,7319
Симметричная схема АФНФ	0,9476	0,8162

На рис. 4 приведены значения метрики *Asym*, которая представляет собой среднеквадратическое значение $\sqrt{\sum_{f=0}^{3}\mu^2(\Delta_{C_i}^f)}$, определяющее степень асимметрии множеств нулевых и единичных ответов всех четырех экземпляров схем АФНФ. Так, полученные значения *Asym* для симметричных схем (0,1937 нс) существенно меньше аналогичного значения (0,4119 нс) для классических АФНФ, что подтверждается вычисленными характеристиками единообразия *Un* (табл. 4).



a) *b)*

Рис. 4. Значения величин задержек $\Delta_{C_i}^f$ для классической схемы АФНФ (*a*) и симметричной схемы АФНФ (*b*)

Fig. 4. Delay values $\Delta_{C_i}^f$ *for classical APUF scheme (a) and symmetrical APUF scheme (b)*

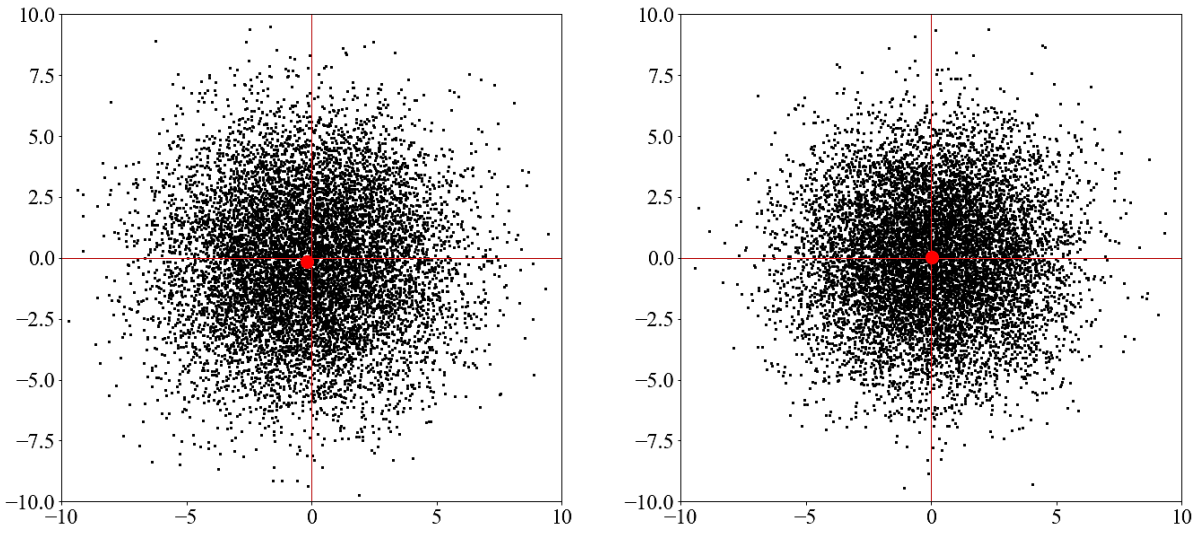


Рис. 5. Графический тест для классической схемы АФНФ (*a*) и симметричной схемы АФНФ (*b*) (*f* = 0)
Fig. 5. Graphical test for classical APUF scheme (a) and symmetrical APUF scheme (b) (f = 0)

Асимметрия множеств значений $\Delta_{C_i}^{f}$ выявляется с помощью графического теста «Распределение на плоскости», результаты которого приведены на рис. 5 и коррелируются со значениями из табл. 3.

Заключение. В статье предложен подход к построению АФНФ, основанный на применении базового элемента, в котором реализованы функции *Generate*, *Select* и *Switch*. Важным отличием С_АФНФ от классических и сбалансированных АФНФ является то, что функции *Generate*, *Select* и *Switch* базового элемента выполняются независимыми его компонентами и задаются разными битами запроса. В отличие от сбалансированных АФНФ С_АФНФ не требуют реализации процедуры балансировки как в процессе изготовления АФНФ, так и при ее применении на практике. Несомненным преимуществом С_АФНФ по отношению к сбалансированным АФНФ, исследованным в работе [24], является достижение симметричности для всего множества запросов C_i. В случае сбалансированных АФНФ симметрия достигается только для сбалансированных запросов C_i, для которых выполняется равенство нулевых и единичных разрядов запроса. Экспериментально подтвержден эффект улучшения характеристик подобных С_АФНФ и в первую очередь их вероятностных свойств, выраженных в равной вероятности ответов, т. е. в отсутствии асимметрии. Перспективным представляется дальнейшее развитие идей построения С_АФНФ, экспериментальное исследование их характеристик, а также анализ устойчивости к различного рода атакам, в том числе и с использованием машинного обучения.

Вклад авторов. *В. Н. Ярмолик* предложил идею построения симметричных физически неклонируемых функций, *А. А. Иванюк* принял участие в обобщении и анализе полученных результатов, а также провел экспериментальные исследования.

Список использованных источников

1. Pappu, R. Physical One-Way Functions: PhD Thesis in Media Arts and Sciences / R. Pappu. – Cambridge : Massachusetts Institute of Technology, 2001. – 154 p.

2. Silicon physical random functions / B. Gassend [et al.] // Proc. of the 9th Computer and Communications Security Conf. (CCS'02), Washington, DC USA, 18–22 Nov. 2002. – Washington, 2002. – P. 148–160.

3. Security with Noisy Data: On Private Biometrics, Secure Key Storage and Anti-Counterfeiting / eds.: P. Tuyls, B. Skoric. – N. Y., USA : Springer, 2007. – 339 p.

4. PUFKY: A fully functional PUF-based cryptographic key generator / R. Maes, A. Van Herrewege, I. Verbauwhede // Proc. of 14th Intern. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2012), Leuven, Belgium, 9–12 Sept. 2012. – Leuven, 2012. – P. 302–319.

5. Robust key extraction from physical uncloneable functions / B. Skoric, P. Tuyls, W. Ophey // Proc. of Intern. Conf. Applied Cryptography and Network Security, N. Y., USA, 7–10 June 2005. – N. Y., 2005. – P. 407–422.

6. Ярмолик, В. Н. Физически неклонируемые функции / В. Н. Ярмолик, Ю. Г. Вашинко // Информатика. – 2011. – № 2(30). – С. 92–103.

7. Suh, G. E. Physical unclonable functions for device authentication and secret key generation / G. E. Suh, S. Devadas // Proc. of Intern. Design Automation Conf., DAC 2007, San Diego, California, USA, 4–8 June 2007. – San Diego, 2007. – P. 9–14.

8. Böhm, C. Physical Unclonable Functions in Theory and Practice / C. Böhm, M. Hofer. – N. Y. : Springer Science + Business Media, 2013. – 270 p.

9. Rührmair, U. Strong PUFs: models, constructions, and security proofs / U. Rührmair, H. Busch, S. Katzenbeisser // Towards Hardware-Intrinsic Security / eds.: A.-R. Sadeghi, D. Naccache. – Berlin, Heidelberg : Springer, 2010. – P. 79–96.

10. A technique to build a secret key in integrated circuits for identification and authentication applications / J. W. Lee [et al.] // Proc. of Intern. Symp. VLSI Circuits (VLSI'04), Honolulu, Hawaii, USA, 7–19 June 2004. – Honolulu, 2004. – P. 176–179.

11. Extracting secret keys from integrated circuits / D. Lim [et al.] // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. – 2005. – Vol. 13, no. 10. – P. 1200–1205.

12. Иванюк, А. А. Физическая криптография и защита цифровых устройств / А. А. Иванюк, С. С. Заливако // Доклады БГУИР. – 2019. – № 2(120). – С. 50–58.

13. Ярмолик, В. Н. Физически неклонируемые функции с управляемой задержкой распространения сигналов / В. Н. Ярмолик, А. А. Иванюк, Н. Н. Шинкевич // Информатика. – 2022. − Т. 19, № 1. – С. 32–49.

14. Using statistical models to improve the reliability of delay-based PUFs / X. Xu, W. Burleson, D. E. Holcomb // Proc. of IEEE Computer Society Annual Symp. on VLSI, Pittsburgh, PA, USA, 11–13 July 2016. – Pittsburgh, 2016. – P. 547–552.

15. An analysis of delay based PUF implementations on FPGA / S. Morozov, A. Maiti, P. Schaumont // Proc. of Intern. Symp. on Applied Reconfigurable Computing: Tools and Applications (ARC 2010), Los Angeles, CA, US, 25–27 Mar. 2010. – Los Angeles, 2010. – P. 382–387.

16. Клыбик, В. П. Метод увеличения стабильности физически неклонируемой функции типа «арбитр» / В. П. Клыбик, С. С. Заливако, А. А. Иванюк // Информатика. – 2017. − № 1(53). – С. 31–43.

17. Ярмолик, В. Н. Физически неклонируемые функции типа арбитр с заведомо асимметричными парами путей / В. Н. Ярмолик, А. А. Иванюк // Доклады БГУИР. – 2022. – № 4(20). – С. 71–79.

18. Secure and reliable XOR arbiter PUF design: An experimental study based on 1 trillion challenge response pair measurements / C. Zhou, K. K. Parhi, C. H. Kim // Proc. of the 54th Annual Design Automation, Austin, TX, USA, 18 June 2017. – Austin, 2017. – P. 18–22.

19. Implementation of double arbiter PUF and its performance evaluation on FPGA / T. Machida [et al.] // Proc. of the 20th Asia and South Pacific Design Automation Conf., Chiba, Japan, 19 Jan. 2015. – Chiba, 2015. – P. 6–7.

20. Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise / J. Delvaux, I. Verbauwhede // Proc. of IEEE Intern. Symp. on Hardware-Oriented Security and Trust (HOST), Austin, TX, USA, 2–3 June 2013. – Austin, 2013. – P. 137–142.

21. PUF modeling attacks on simulated and silicon data / U. Rührmair [et al.] // IEEE Transactions on Information Forensics and Security. – 2013. – Vol. 11, no. 8. – P. 1876–1891.

22. Шамына, А. Ю. Построение и балансировка путей физически неклонируемой функции типа арбитр на FPGA / А. Ю. Шамына, А. А. Иванюк // Информатика. – 2022. – Т. 19, № 4. – С. 27–41.

23. Ярмолик, В. Н. Двухмерные физически неклонируемые функции типа арбитр / В. Н. Ярмолик, А. А. Иванюк // Информатика. – 2023. – Т. 20, № 1. – С. 7–26.

24. Ярмолик, В. Н. Сбалансированные физически неклонируемые функции типа арбитр / В. Н. Ярмолик, А. А. Иванюк // Безопасность информационных технологий. – 2023. – № 1(30). – С. 92–107.

25. Ярмолик, В. Н. Контроль и диагностика вычислительных систем / В. Н. Ярмолик. – Минск : Бестпринт, 2019. – 387 с.

26. Implementation of pseudo-linear feedback shift register-based physical unclonable functions on silicon and sufficient Challenge-Response pair acquisition using Built-In Self-Test before shipping / Y. Ogasahara [et al.] // Integration, the VLSI J. – 2020. – Vol. 71. – P. 144–153.

27. Evaluation of physical unclonable functions for 28-nm process field-programmable gate arrays / Y. Hori [et al.] // J. of Information Processing. – 2014. – Vol. 22, no. 2. – P. 344–356.

References

1. Pappu R. *Physical One-Way Functions: PhD Thesis in Media Arts and Sciences*. Cambridge, Massachusetts Institute of Technology, 2001, 154 p.

2. Gassend B., Clarke D., Dijk M. S., Devadas S. Silicon physical random functions. *Proceedings of the 9th Computer and Communications Security Conference (CCS'02), Washington, DC USA, 18–22 November 2002.* Washington, 2002, pp. 148–160.

3. Tuyls P., Skoric B. (eds.). *Security with Noisy Data: On Private Biometrics, Secure Key Storage and Anti-Counterfeiting*. New York, USA, Springer, 2007, 339 p.

4. Maes R., Van Herrewege A., Verbauwhede I. PUFKY: A fully functional PUF-based cryptographic key generator. *Proceedings of 14th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2012), Leuven, Belgium, 9–12 September 2012.* Leuven, 2012, pp. 302–319.

5. Skoric B., Tuyls P., Ophey W. Robust key extraction from physical uncloneable functions. *Proceedings of International Conference Applied Cryptography and Network Security, New York, USA, 7–10 June 2005*. New York, 2005, pp. 407–422.

6. Yarmolik V. N., Vashinko Y. G. *Physical unclonable functions*. Informatika [*Informatics*], 2011, no. 2(30), pp. 92–103 (In Russ.).

7. Suh G. E., Devadas S. Physical unclonable functions for device authentication and secret key generation. *Proceedings of International Design Automation Conference, DAC 2007, San Diego, California, USA, 4–8 June 2007*. San Diego, 2007, pp. 9–14.

8. Böhm C., Hofer M. *Physical Unclonable Functions in Theory and Practice*. New York, Springer Science + Business Media, 2013, 270 p.

9. Rührmair U., Busch H., Katzenbeisser S. Strong PUFs: models, constructions, and security proofs. *Towards Hardware-Intrinsic Security*. In A.-R. Sadeghi, D. Naccache (eds.). Berlin, Heidelberg, Springer, 2010, pp. 79–96.

10. Lee J. W., Lim D., Gassend B., Suh G. E., Van Dijk M., Devadas S. A technique to build a secret key in integrated circuits for identification and authentication applications. *Proceedings of International Symposium VLSI Circuits (VLSI'04), Honolulu, Hawaii, USA, 7–19 June 2004.* Honolulu, 2004, pp. 176–179.

11. Lim D., Lee J. W., Gassend B., Suh G. E., Van Dijk M., Devadas S. Extracting secret keys from integrated circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2005, vol. 13, no. 10, pp. 1200–1205.

12. Ivaniuk A. A., Zalivaka S. S. *Physical cryptography and security of digital devices.* Doklady Belorusskogo gosudarstvennogo universiteta informatiki i radioèlektroniki [*Reports of the Belarusian State University of Informatics and Radioelectronics*], 2019, no. 2(120), pp. 50–58 (In Russ.).

13. Yarmolik V. N., Ivaniuk A. A., Shynkevich N. N. *Physically unclonable functions with controlled propagation delay.* Informatika [*Informatics*], 2022, vol. 19, no. 1, pp. 32–49 (In Russ.).

14. Xu X., Burleson W., Holcomb D. E. Using statistical models to improve the reliability of delay-based PUFs. *Proceedings of IEEE Computer Society Annual Symposium on VLSI, Pittsburgh, PA, USA, 11–13 July 2016.* Pittsburgh, 2016, pp. 547–552.

15. Morozov S., Maiti A., Schaumont P. An analysis of delay based PUF implementations on FPGA. *Proceedings of International Symposium on Applied Reconfigurable Computing: Tools and Applications (ARC 2010), Los Angeles, CA, US, 25–27 March 2010.* Los Angeles, 2010, pp. 382–387.

16. Klybik V. P., Zalivaka S. S., Ivaniuk A. A. *Reliability enhancement method for "arbiter" physicaly unclonable function*. Informatika [*Informatics*], 2017, no. 1(53), pp. 31–43 (In Russ.).

17. Yarmolik V. N., Ivaniuk A. A. *Arbiter physical unclonable functions with asymmetric pairs of paths*. Doklady Belorusskogo gosudarstvennogo universiteta informatiki i radioèlektroniki [*Reports of the Belarusian State University of Informatics and Radioelectronics*], 2022, no. 4(20), pp. 71–79 (In Russ.).

18. Zhou C., Parhi K. K., Kim C. H. Secure and reliable XOR arbiter PUF design: An experimental study based on 1 trillion challenge response pair measurements. *Proceedings of the 54th Annual Design Automation, Austin, TX, USA, 18 June 2017*. Austin, 2017, pp. 18–22.

19. Machida T., Yamamoto D., Iwamoto M., Sakiyama K. Implementation of double arbiter PUF and its performance evaluation on FPGA. *Proceedings of the 20th Asia and South Pacific Design Automation Conference, Chiba, Japan, 19 January 2015*. Chiba, 2015, pp. 6–7.

20. Delvaux J., Verbauwhede I. Side channel modeling attacks on 65nm arbiter PUFs exploiting CMOS device noise. *Proceedings of IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Austin, TX, USA, 2–3 June 2013*. Austin, 2005, pp. 137–142.

21. Rührmair U., Sölter J., Sehnke F., Xu X., Mahmoud A., …, Devadas S. PUF modeling attacks on simulated and silicon data. *IEEE Transactions on Information Forensics and Security*, 2013, vol. 11, no. 8, pp. 1876–1891.

22. Shamyna A. Yu., Ivaniuk A. A. *Creating and balancing the paths of arbiter-based physically unclonable functions on FPGA*. Informatika [*Informatics*], 2022, vol. 19, no. 4, pp. 27–41 (In Russ.).

23. Yarmolik V. N., Ivaniuk A. A. *2D physically unclonable functions of the arbiter type*. Informatika [*Informatics*], 2023, vol. 20, no. 1, pp. 7–26 (In Russ.).

24. Yarmolik V. N., Ivaniuk A. A. *Balanced arbiter physical uncloneable functions*. Bezopasnost' informacionnyh tehnologij [*IT Security*], 2023, no. 1(30), pp. 92–107.

25. Yarmolik V. N. Kontrol' i diagnostika vuchislitel'nuh system. *Monitoring and Diagnostics of Computer Systems*. Minsk, Bestprint, 2019, 387 p. (In Russ.).

26. Ogasahara Y., Hori Y., Katashita T., Iizuka T., Awano H., …, Koike H. Implementation of pseudo-linear feedback shift register-based physical unclonable functions on silicon and sufficient Challenge-Response pair acquisition using Built-In Self-Test before shipping. *Integration, the VLSI Journal*, 2020, vol. 71, pp. 144–153.

27. Hori Y., Kang H., Katashita T., Satoh A., Kawamura S., Kobara K. Evaluation of physical unclonable functions for 28-nm process field-programmable gate arrays. *Journal of Information Processing*, 2014, vol. 22, no. 2, pp. 344–356.

Информация об авторах

Ярмолик Вячеслав Николаевич, доктор технических наук, профессор, Белорусский государственный университет информатики и радиоэлектроники.
E-mail: yarmolik10ru@yahoo.com

Иванюк Александр Александрович, доктор технических наук, доцент, профессор кафедры информатики, Белорусский государственный университет информатики и радиоэлектроники.
E-mail: ivaniuk@bsuir.by

Information about the authors

Vyacheslav N. Yarmolik, D. Sc. (Eng.), Prof., Belarusian State University of Informatics and Radioelectronics.
E-mail: yarmolik10ru@yahoo.com

Alexander A. Ivaniuk, D. Sc. (Eng.), Assoc. Prof., Prof. of Computer Science Department, Belarusian State University of Informatics and Radioelectronics.
E-mail: ivaniuk@bsuir.by