ЗАЩИТА ИНФОРМАЦИИ И НАДЕЖНОСТЬ СИСТЕМ
INFORMATION PROTECTION AND SYSTEM RELIABILITY

УДК 004.832.32
https://doi.org/10.37661/1816-0301-2023-20-1-7-26

Оригинальная статья
Original Paper

Двухмерные 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), основанного на различии задержек по входам многочисленных модификаций базового элемента путем увеличения как количества входов, так и топологии их подключения. Подобный подход позволяет строить двухмерные физически неклонируемые функции (2D-АФНФ), в которых в отличие от классических АФНФ запрос, формируемый для каждого базового элемента, выбирает пару путей не из двух возможных, а из большего их количества. Актуальность данного исследования связана с активным развитием физической криптографии. В работе преследуются следующие цели: построение базовых элементов АФНФ и их модификаций, разработка методики построения 2D-АФНФ.

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

Результаты. Показано, что в классических АФНФ применяется стандартный базовый элемент, выполняющий две функции, а именно функцию выбора пары путей *Select* и функцию переключения путей *Switch*, которые за счет их совместного использования позволяют достичь высоких характеристик. В первую очередь это касается стабильности функционирования АФНФ, характеризующейся небольшим числом запросов, для которых ответ случайным образом принимает одно из двух возможных значений: 0 или 1. Предложены модификации базового элемента в части реализаций его функций *Select* и *Switch*. Приводятся новые структуры базового элемента с внесенными модификациями их реализаций, в том числе в части увеличения количества пар путей базового элемента, из которых путем запроса выбирается одна из них и конфигурации их переключений. Применение разнообразных базовых элементов позволяет улучшать основные характеристики АФНФ, а также нарушать регулярность их структуры, которая является главной причиной взлома АФНФ путем машинного обучения.

Заключение. Предложенный подход к построению 2D-АФНФ, основанный на различии задержек сигналов через базовый элемент, показал свою работоспособность и перспективность. Экспериментально подтвержден эффект улучшения характеристик подобных ФНФ, и в первую очередь стабильности их функционирования. Перспективным представляется дальнейшее развитие идеи построения 2D-АФНФ, экспериментальное исследование их характеристик и устойчивости к различного рода атакам, в том числе с использованием машинного обучения.

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

Благодарности. Авторы выражают искреннюю благодарность резиденту ПВТ компании «СК хайникс мемори солюшенс Восточная Европа» за предоставленное оборудование для проведения экспериментов в рамках работы совместной учебной лаборатории с Белорусским государственным университетом информатики и радиоэлектроники.

Для цитирования. Ярмолик, В. Н. Двухмерные физически неклонируемые функции типа арбитр / В. Н. Ярмолик, А. А. Иванюк // Информатика. – 2023. − Т. 20, № 1. – С. 7–26.
https://doi.org/10.37661/1816-0301-2023-20-1-7-26

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию | Received 11.10.2022
Подписана в печать | Accepted 21.11.2022
Опубликована | Published 29.03.2023

2D 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) is being solved, based on the difference in delay times for the inputs of numerous modifications of the base element, due to both an increase in the number of inputs and the topology of their connection. Such an approach allows building two-dimensional physically unclonable functions (2D-APUF), in which, unlike classical APUF, the challenge generated for each basic element selects a pair of paths not from two possible, but from a larger number of them. The relevance of such a study is associated with the active development of physical cryptography. The following goals are pursued in the work: the construction of the basic elements of the APUF and their modifications, the development of a methodology for constructing 2D-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 is shown that the classical APUF uses a standard basic element that performs two functions, namely, the function of choosing a pair of paths *Select* and the function of switching paths *Switch*, which, due to their joint use, allow achieving high performance. First of all, this concerns the stability of the APUF functioning, which is characterized by a small number of challenge, for which the response randomly takes one of two possible values 0 or 1. Modifications of the base element in terms of the implementations of its *Select* and *Switch* functions are proposed. New structures of the base element are presented in which the modifications of their implementations are made, including in terms of increasing the number of pairs of paths of the base element from which one of them is selected by the challenge, and the configurations of their switching. The use of various basic elements makes it possible to improve the main characteristics of APUF, as well as to break the regularity of their structure, which was the main reason for hacking APUF through machine learning.

Conclusion. The proposed approach to the construction of physically unclonable 2D-APUF functions, based on the difference in signal delays through the base element, has shown its efficiency and promise. The effect of improving the characteristics of such PUFs has been experimentally confirmed with noticeable improvement in the stability of their functioning. It seems promising to further develop the ideas of constructing two-dimensional physically unclonable functions of the arbiter type, as well as experimental study of their characteristics, as well as resistance to various types of attacks, including using machine learning.

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

Acknowledgements. The authors express their sincere gratitude to the HTP resident of the "SK hynix memory solutions Eastern Europe" company for the equipment provided for carrying out experiments within the framework of the joint laboratory with the Belarusian State University of Informatics and Radioelectronics.

For citation. 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.). https://doi.org/10.37661/1816-0301-2023-20-1-7-26

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

Введение. Физически неклонируемые функции (Physical Unclonable Functions, PUF) предлагают многообещающие решения для настоящих и будущих задач информационной безопасности [1, 2]. Изначально PUF предназначались для защиты от появления поддельных нелегальных цифровых устройств; по сути, – для защиты авторских прав на цифровые устройства [3, 4]. В настоящее время сфера применения PUF значительно расширилась за счет их активного применения в криптографии для целей идентификации, аутентификации, генерирования криптографических ключей, а также реализации различных криптографических протоколов [5–8].

Первоначально PUF назывались физические однонаправленные функции (Physical One-Way Functions, POWF) или физические случайные функции (Physical Random Functions, PRF) [1, 2]. Несмотря на то что последние два определения были сформулированы исторически раньше, в настоящее время в основном употребляется название PUF, которое в русскоязычной литературе представляется как *физически неклонируемые функции* (ФНФ) [9–11]. Идея ФНФ впервые была представлена Р. Паппу (R. Pappu) в его пионерской работе [1], где он впервые сделал попытку сформулировать основные понятия и определения в данной области.

Одно из наиболее широко используемых на сегодняшний день определений ФНФ было предложено П. Туилсом (P. Tuyls) [3] как обобщение и развитие работ Р. Паппу. Физически неклонируемые функции, по его определению, – это физические системы, неотъемлемым свойством которых является неклонируемость, т. е. невозможность воспроизведения двух идентичных ФНФ. У таких систем свойство неклонируемости обусловлено тем фактом, что они состоят из множества компонентов, параметры которых в процессе создания подобных физических систем принимают случайные значения. Наличие компонентов со случайными величинами их параметров и характеристик, а также невозможность контролировать и управлять этими параметрами элементов ФНФ во время производства делают их уникальными и физически неклонируемыми. ФНФ описываются значениями входных и соответствующих им 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). В простейшем случае ФНФ можно рассматривать как функцию $R = F(C)$, которая преобразует запросы C в ответы R [3, 9].

Исчерпывающее определение ФНФ как системы со сверхбольшим объемом информации (Super High Information Content, SHIC) было предложено У. Рухрмаером (U. Rührmair) [9, 12].

Физически неклонируемые функции представляют собой сложные неупорядоченные физические системы с чрезвычайно большим объемом структурной информации, которые удовлетворяют следующим требованиям:

1. Структурная информация подобных систем может быть извлечена надежно и многократно путем проведения измерений для различных запросов C и получения ответов R.

2. Количество возможных запросов C должно быть настолько велико, что значения всех соответствующих ответов R не могут быть получены путем перебора всех возможных запросов C за реальный временной промежуток.

3. Ввиду наличия в системе чрезвычайно большого объема структурной информации должно быть невозможным моделировать, рассчитывать, эмулировать или каким-либо другим математическим способом предсказывать пару «запрос-ответ» (C_j, R_j), зная другую пару (C_i, R_i) или некоторое множество таких пар.

4. Для физической системы с чрезвычайно большим объемом структурной информации должно быть чрезвычайно сложным ее физическое воспроизведение или клонирование (повторение) как аналогичной физической системы, описываемой идентичным множеством пар «запрос-ответ».

Основные проблемы при создании ФНФ заключаются в противоречии первого требования, которое характеризует стабильность их функционирования, и третьего требования о непредсказуемости, случайности таких функций. Попытка повысить стабильность ФНФ увеличивает их уязвимость для различного рода атак, в особенности с применением современных достижений машинного обучения [13–15].

Для цифровых устройств фундаментальным подходом при реализации большинства разновидностей ФНФ является создание такого цифрового устройства ФНФ, выходное значение которого определяется случайными значениями временных параметров, чаще всего задержек электрических сигналов. Из-за технологических вариаций во время изготовления произвольного цифрового устройства время задержки сигналов по определенному его пути будет незначительно меняться от цифрового устройства к цифровому устройству и от кристалла к кристаллу, несмотря на идентичность их топологии и функциональности [16–18].

Исторически первой ФНФ является АФНФ (Arbiter PUF), предложенная в 2002 г. [2]. АФНФ оказалась удачным решением, основанным на различии задержек прохождения сигнала через цифровые элементы. При этом различие задержек стало основным фактором, влияющим на свойства подобной ФНФ.

Развивая идею различия задержки прохождения сигнала через элемент, в настоящей работе предлагается использовать для построения новых структур АФНФ отличия задержек сигнала по входам цифрового элемента. Показывается, что как сам цифровой элемент, так и каждый его вход уникальны и неповторимы в части задержки прохождения сигнала на выходе элемента. В статье предлагаются новые структурные решения для базового элемента АФНФ, характеризующиеся отличающимся набором значений временных задержек и их количеством, что позволило улучшить основные характеристики АФНФ, и в первую очередь стабильность ее функционирования. Впервые рассматривается идея построения 2D-АФНФ (2D-APUF).

1. ФНФ типа арбитр. В общем случае при реализации АФНФ изготавливаются два функционально и топологически идентичные электрические пути, представляющие собой последовательно подключенные элементы и их межсоединения. Очевидно, что оба пути будут иметь близкие значения величин задержек распространения по ним сигналов, однако они будут принципиально разными в силу технологических вариаций в процессе производства. Процедура измерения времени распространения сигналов заключается в одновременной подаче на входы обоих путей сигнала и определении того из них, на выходе которого сигнал появится быстрее. Пары симметричных путей задержки электрического сигнала изготавливаются таким образом, что одновременно из большого множества пар выбирается одна пара за счет формирования определенного запроса *C*. Далее для выбранной пары путей определяется, какой из них более быстрый, и формируется ответ *R*. Непредсказуемость и случайность ответов для конкретной реализации АФНФ служат фундаментальной основой подобных функций. В то же время каждая реализация АФНФ обеспечивает стабильность, т. е. повторяемость ответов на одни и те же запросы, которые не повторяют их значения для других реализаций функционально и топологически идентичных АФНФ.

Классическая схема АФНФ (рис. 1) [6] строится с использованием n последовательно подключенных пар двухвходовых мультиплексоров (*MUX*). Адресные входы (*Adr*) обоих мультиплексоров MUX_{1j} и MUX_{2j} каждой пары объединяются и применяются в качестве одного из входов для задания значения одного бита запроса c_j. Запросом в данном случае выступает n-разрядный двоичный вектор $C_i = c_0\, c_1\, c_2 \ldots c_{n-1}$, где $c_j \in \{0, 1\}$, $j \in \{0, 1, 2, \ldots, n-1\}$. Запрос C_i в схеме АФНФ формирует два пути таким образом, что если $c_j = 0$ для j-й ступени АФНФ, то для построения первого пути используется верхний мультиплексор MUX_{1j}, а для построения второго пути – нижний MUX_{2j}; если $c_j = 1$, то наоборот. Каждая пара путей имеет общий вход, а выходы первого и второго путей подключены соответственно к входу D D-триггера и его синхронизирующему входу *Clk*. В данном случае D-триггер является арбитром и перед проведением эксперимента устанавливается в исходное нулевое состояние. Для конкретного запроса C_i конфигурируется уникальная пара путей и генерируется ответ $R_i \in \{0, 1\}$ как результат эксперимента по определению того, по какому из путей (первому или второму) задержка входного

импульсного сигнала меньше. Например, если по первому пути, то $R_i = 1$, а если по второму, то $R_i = 0$. Очевидно, что количество пар путей с увеличением n растет экспоненциально и равняется 2^n.

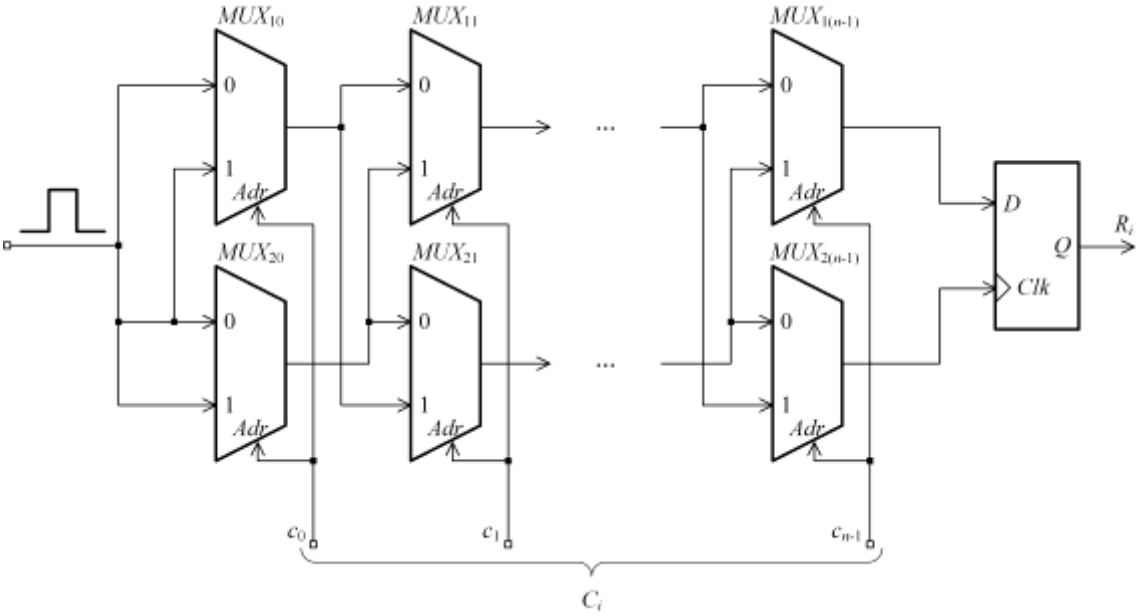

Рис. 1. АФНФ на базе двухвходовых мультиплексоров

Fig. 1. APUF based on two-input multiplexers

Все известные решения построения ФНФ, в том числе и АФНФ, основаны на том, что задержка по конкретному пути (элементу) имеет случайное значение, определяемое множеством факторов, которые влияют на ее величину в процессе производства ФНФ. Однако у реальных ФНФ эти случайные задержки имеют неизменное и неуправляемое значение, исключая влияние внешних факторов (температуры, давления, электромагнитного излучения и др.) и временной деградации. Их неизменность, с одной стороны, обусловливает стабильность функционирования ФНФ, а с другой – открывает возможности для различного рода атак на ФНФ по предсказанию либо описанию их поведения.

Базируясь исключительно на детерминированном поведении АФНФ, чаще всего и строят различные их математические модели. Наиболее распространенная из них основана на том, что каждая j-я ступень АФНФ, состоящая из пары мультиплексоров MUX_{1j} и MUX_{2j}, может быть описана двумя параметрами, а именно разностями задержек $\delta_{0,j}$ и $\delta_{1,j}$. Разность задержки $\delta_{0,j}$ для j-й ступени АФНФ определяется как добавленная разность задержек прохождения сигнала по двум путям через MUX_{1j} и MUX_{2j} при $c_j = 0$, а разность задержки $\delta_{1,j}$ – при $c_j = 1$. Если эти два параметра известны для каждой ступени АФНФ, окончательную разницу задержки для каждой пары путей можно легко определить, если учесть возможный эффект переключения на каждой ступени. Переключение одного пути на j-й ступени АФНФ с MUX_{1j} на MUX_{2j}, а другого пути с MUX_{2j} на MUX_{1j} эквивалентно изменению знака разницы задержек сигналов пары путей на предыдущих ступенях АФНФ. Соответственно, разница задержки d_j после j-й ступени может вычисляться рекурсивно:

$$d_j = d_{j-1} \cdot (-1)^{c_j} + \delta_{c_j j}. \quad (1)$$

Анализ соотношения (1) показывает, что ответ R_i на запрос C_i для АФНФ, представленной на рис. 1, определяется знаком разницы задержек d_{n-1} импульсного сигнала по выбранным путям в соответствии с запросом C_i. В рамках подобных моделей описания АФНФ определяющим фактором являются величины добавленной разности задержек $\delta_{0,j}$ и $\delta_{1,j}$, от которых зависят

основные свойства и характеристики АФНФ. В свою очередь, эти величины для *j*-й ступени АФНФ (рис. 1) вычисляются согласно соотношениям

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

Численное значение $\Delta(0)_{1,j}$ определяет временную задержку прохождения сигнала с нулевого входа, обозначенного символом 0, на первом мультиплексоре (MUX_{1j}) *j*-й ступени АФНФ на его выход, а значение $\Delta(0)_{2,j}$ – задержку на втором мультиплексоре (MUX_{2j}). Величины $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$ представляют собой задержки сигналов по единичным входам соответствующих мультиплексоров. Отметим, что все четыре значения, а именно $\Delta(0)_{1,j}$, $\Delta(0)_{2,j}$, $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$, и являются источниками непредсказуемости поведения АФНФ (см. рис. 1). Все четыре величины принимают случайные значения в результате влияния множества случайных факторов на процесс изготовления АФНФ и конкретно *j*-й его ступени. При функционировании АФНФ эти величины в идеальном случае имеют неизменные значения и участвуют в определении величин добавленной разности задержек $\delta_{0,j}$ и $\delta_{1,j}$ согласно соотношениям (2). Под неизменностью указанных величин понимаются такие изменения их значений, которые не нарушают повторяемость ответов для одного и того же запроса. Далее результат R_i функционирования АФНФ при подаче запроса C_i определяется комбинацией (1) величин разности задержек $\delta_{c_j,j}$. Таким образом, для получения ответа используются только две случайные величины $\delta_{0,j}$ и $\delta_{1,j}$, каждая из которых представляет собой разность двух из четырех исходных, предварительно сгенерированных случайных значений $\Delta(0)_{1,j}$, $\Delta(0)_{2,j}$, $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$.

Для АФНФ, представленной на рис. 1, так же, как и для большинства подобных известных решений, базовым элементом является схема, которая состоит из двух двухвходовых мультиплексоров MUX_{1j} и MUX_{2j}. Копирование таких последовательно соединенных схем и лежит в основе создания АФНФ. Отметим, что в процессе производства *j*-го базового элемента формируются фиксированные значения задержек сигнала по всем его четырем входам. В силу различного рода случайных факторов величины этих задержек $\Delta(0)_{1,j}$, $\Delta(0)_{2,j}$, $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$ принимают случайные значения. В идеальном случае указанные задержки сохраняют свои значения в процессе функционирования АФНФ. На этой гипотезе для обеспечения стабильности и повторяемости ответов на генерируемые запросы и основан принцип их функционирования. От сочетания значений задержек $\Delta(0)_{1,j}$, $\Delta(0)_{2,j}$, $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$ зависят основные свойства АФНФ, которые следуют из их математического описания (1). Указанная модель позволяет убедиться в эффективности классической структуры АФНФ при различных сочетаниях задержек. Даже в случае весьма крайних и маловероятных соотношений задержек $\Delta(0)_{1,j}$, $\Delta(0)_{2,j}$, $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$ существуют множества запросов, обеспечивающих устойчивое функционирование АФНФ.

В качестве примера рассмотрим классическую АФНФ для $n = 4$ (рис. 2).

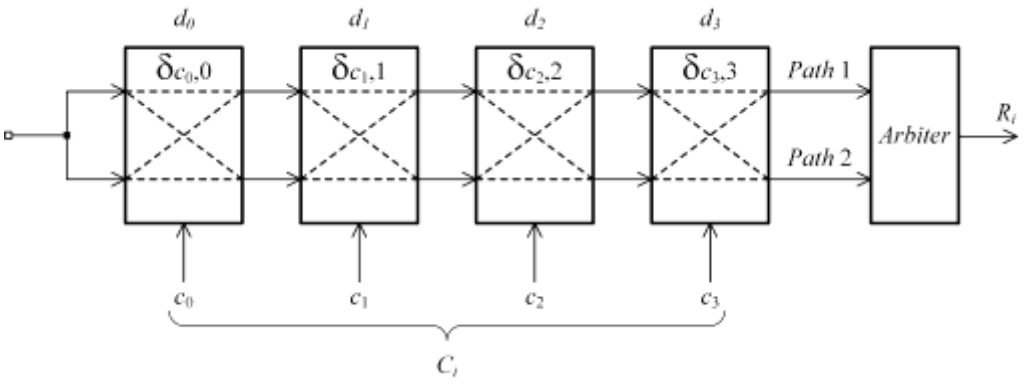

Рис. 2. АФНФ для $n = 4$

Fig. 2. APUF for n = 4

На каждой j-й, $j \in \{0, 1, 2, 3\}$, ступени АФНФ в зависимости от значения c_j запроса C_i формируются задержки распространения сигнала по путям *Path* 1 и *Path* 2 выбранной запросом пары путей (см. рис. 2). Соотношение этих задержек на каждой ступени определяется величиной их добавленной разности $\delta_{c_j, j}$, а значение разности задержки сигнала по пути *Path* 2 по отношению к задержке по пути *Path* 1 – величиной d_j (1). Знак плюс либо минус финального значения d_3 разности задержек и определяет значение ответа $R_i \in \{0, 1\}$.

В качестве двух реализаций АФНФ, показанной на рис. 2, рассмотрим АФНФ$_1$ и АФНФ$_2$, имеющие фиксированные задержки сигналов $\Delta(0)_{1,j}$, $\Delta(0)_{2,j}$, $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$ и соответствующие им величины добавленной разности задержек $\delta_{0,j}$ и $\delta_{1,j}$. В табл. 1 приведены соотношения задержек $\Delta(0)_{1,j}$, $\Delta(0)_{2,j}$, $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$ для обеих реализаций и численные значения добавленной разности задержек $\delta_{0,j}$ и $\delta_{1,j}$ в условных единицах, кратных абстрактной величине задержки d. Первая реализация АФНФ$_1$ в упрощенном виде представляет собой аналог стандартной реализации АФНФ, когда разности задержек каждой ступени принимают произвольные значения. Из табл. 1 видно, что абсолютные величины задержек разнятся так же, как и знаки этих разностей, принимающие оба значения – и плюс, и минус.

Таблица 1
Значения задержек сигналов для АФНФ$_1$ и АФНФ$_2$

Table 1
Signal delay value for $APUF_1$ and $APUF_2$

АФНФ$_1$ *$APUF_1$*	$\Delta(0)_{1,0} > \Delta(0)_{2,0}$	$\Delta(0)_{1,1} > \Delta(0)_{2,1}$	$\Delta(0)_{1,2} > \Delta(0)_{2,2}$	$\Delta(0)_{1,3} < \Delta(0)_{2,3}$
	$\delta_{0,0} = d$	$\delta_{0,1} = d$	$\delta_{0,2} = 2d$	$\delta_{0,3} = -d$
	$\Delta(1)_{1,0} < \Delta(1)_{2,0}$	$\Delta(1)_{1,1} > \Delta(1)_{2,1}$	$\Delta(1)_{1,2} < \Delta(1)_{2,2}$	$\Delta(1)_{1,3} > \Delta(1)_{2,3}$
	$\delta_{1,0} = -d$	$\delta_{1,1} = 2d$	$\delta_{1,2} = -2d$	$\delta_{1,3} = 2d$
АФНФ$_2$ *$APUF_2$*	$\Delta(0)_{1,0} > \Delta(0)_{2,0}$	$\Delta(0)_{1,1} > \Delta(0)_{2,1}$	$\Delta(0)_{1,2} > \Delta(0)_{2,2}$	$\Delta(0)_{1,3} > \Delta(0)_{2,3}$
	$\delta_{0,0} = d$	$\delta_{0,1} = d$	$\delta_{0,2} = d$	$\delta_{0,3} = d$
	$\Delta(1)_{1,0} > \Delta(1)_{2,0}$	$\Delta(1)_{1,1} > \Delta(1)_{2,1}$	$\Delta(1)_{1,2} > \Delta(1)_{2,2}$	$\Delta(1)_{1,3} > \Delta(1)_{2,3}$
	$\delta_{1,0} = d$	$\delta_{1,1} = d$	$\delta_{1,2} = d$	$\delta_{1,3} = d$

Функция АФНФ$_2$ приведена в качестве примера весьма неудачного, аномального случая синтеза АФНФ. АФНФ$_2$ является результатом изготовления АФНФ, когда из-за вариаций производственного процесса задержки $\Delta(0)_{1,j}$ и $\Delta(1)_{1,j}$ мультиплексоров MUX_{1j} всех $n = 4$ ступеней АФНФ$_2$ оказались больше задержек мультиплексоров MUX_{2j}. Следует отметить реальность такой ситуации в технологических процессах изготовления подобных функций, особенно при реализации АФНФ на программируемых структурах [19–21]. Соответственно, все величины добавленной разности задержек $\delta_{0,j}$ и $\delta_{1,j}$ согласно (2) примут положительные значения. Еще более усугубляя этот аномальный случай для АФНФ, предположим, что для всех четырех ступеней АФНФ$_2$ $\delta_{0,j} = \delta_{1,j} = d$.

Полное описание функционирования АФНФ$_1$ и АФНФ$_2$ для $n = 4$ и определенных для них величин добавленной разности задержек $\delta_{0,j}$ и $\delta_{1,j}$ дано в табл. 2. Символ X в табл. 2 означает метастабильное состояние АФНФ, которое относится к нежелательному ее поведению [17, 22]. Этот случай имеет место для запросов C_i, ответом на которые является равенство нулю задержки d_3. Например, для АФНФ$_1$ и запроса $C_9 = c_0\, c_1\, c_2\, c_3 = 1\ 0\ 0\ 1$ распространение сигнала по двум путям *Path* 1 и *Path* 2 в терминах разницы задержки d_i (1) описывается следующими преобразованиями:

$$d_0 = 0 \cdot (-1)^1 + (-d) = -d;\ d_1 = (-d) \cdot (-1)^0 + d = 0;\ d_2 = 0 \cdot (-1)^0 + 2d = 2d;\ d_3 = (2d) \cdot (-1)^1 + 2d = 0.$$

Равенство нулю значения d_3 приводит к тому, что при повторении запроса $C_i = 1\ 0\ 0\ 1$ АФНФ$_1$ будет генерировать случайное значение ответа $R_i \in \{0, 1\}$. Это и есть метастабильное

поведение АФНФ$_2$. На практике метастабильное состояние и соответствующие ему запросы C_i являются весьма нежелательными, так как нарушают стабильность функционирования АФНФ [17, 22].

Таблица 2
Описание функционирования АФНФ$_1$ и АФНФ$_2$
Table 2
Description of functioning APUF$_1$ and APUF$_2$

i	C_i				АФНФ$_1$ *APUF$_1$*					АФНФ$_2$ *APUF$_2$*				
					d_j				R_i	d_j				R_i
	c_0	c_1	c_2	c_3	d_0	d_1	d_2	d_3		d_0	d_1	d_2	d_3	
0	0	0	0	0	d	$2d$	$4d$	$3d$	1	d	$2d$	$3d$	$4d$	1
1	0	0	0	1	d	$2d$	$4d$	$-2d$	0	d	$2d$	$3d$	$-2d$	0
2	0	0	1	0	d	$2d$	$-4d$	$-5d$	0	d	$2d$	$-d$	0	X
3	0	0	1	1	d	$2d$	$-4d$	$6d$	1	d	$2d$	$-d$	$2d$	1
4	0	1	0	0	d	d	$3d$	$2d$	1	d	0	d	$2d$	1
5	0	1	0	1	d	d	$3d$	$-d$	0	d	0	d	0	X
6	0	1	1	0	d	d	$-3d$	$-4d$	1	d	0	d	$2d$	1
7	0	1	1	1	d	d	$-3d$	$5d$	0	d	0	d	0	X
8	1	0	0	0	$-d$	0	$2d$	d	1	d	$2d$	$3d$	$4d$	1
9	1	0	0	1	$-d$	0	$2d$	0	X	d	$2d$	$3d$	$-2d$	0
10	1	0	1	0	$-d$	0	$-2d$	$-3d$	0	d	$2d$	$-d$	0	X
11	1	0	1	1	$-d$	0	$-2d$	$4d$	1	d	$2d$	$-d$	$2d$	1
12	1	1	0	0	$-d$	$3d$	$5d$	$4d$	1	d	0	d	$2d$	1
13	1	1	0	1	$-d$	$3d$	$5d$	$-3d$	0	d	0	d	0	X
14	1	1	1	0	$-d$	$3d$	$-5d$	$-6d$	0	d	0	d	$2d$	1
15	1	1	1	1	$-d$	$3d$	$-5d$	$7d$	1	d	0	d	0	X

Главный вывод, который можно сделать в результате анализа данных в табл. 2, касается высокой эффективности АФНФ за счет того, что базовый элемент выполняет в том числе и функцию переключения *Switch* путей *j*-й ступени АФНФ, одного с MUX_{1j} на MUX_{2j}, а другого с MUX_{2j} на MUX_{1j}. Переключение путей позволяет нивелировать асимметрию задержек двух путей, вызванную аномальными отличиями характеристик элементов, которые реализуют АФНФ. Этот факт подтверждается примером АФНФ$_2$, которая, несмотря на детерминированное отклонение задержек ее элементов, в принципе может рассматриваться в качестве рабочей версии АФНФ. Отметим, что реализация АФНФ$_2$ по той же методологии выбора пары путей, но на основе базового элемента без функции переключения путей, как в работе [23] на базе буфера с тремя состояниями, привела бы к созданию неработоспособной версии АФНФ. Реализация АФНФ$_2$ с использованием двух раздельных множеств путей, когда первый путь пары строится из первого множества элементов, а второй путь – из второго множества, привела бы к получению неизменного ответа на любой запрос АФНФ$_2$.

Наличие различного рода асимметричных аномалий в большей мере присуще реализациям АФНФ на программируемой логике типа FPGA. Показано, что асимметрия задержки АФНФ, реализованной на FPGA, из-за асимметрии маршрутизации более чем в 10 раз выше, чем случайная ее вариация из-за производственного процесса [19]. Однако, по мнению авторов, большинство существующих реализаций АФНФ на FPGA позволяет достичь неплохих результатов только за счет удачного выбора базового элемента, состоящего из двух мультиплексоров, и в большей мере за счет его функции переключения путей.

В то же время однородность структуры АФНФ, состоящей из последовательно подключенных одинаковых базовых элементов, является ее основным недостатком и позволяет проводить различного рода атаки по их взлому [13, 14].

2. Модификации базового элемента АФНФ. Классическая структура АФНФ строится на основе базового элемента из двух двухвходовых мультиплексоров MUX_{1j} и MUX_{2j}, подробно описанного в разд. 1. Базовый элемент имеет два входа и два выхода, что позволяет путем последовательного подключения n подобных элементов строить основную структуру АФНФ, отвечающую за выбор пары путей из 2^n возможных пар. Уникальность такого базового элемента заключается в простоте его аппаратной реализации и достаточно высокой эффективности, позволяющей создавать как классические схемы АФНФ [24, 25], так и различные их модификации, в том числе на программируемых структурах типа FPGA [19–21]. Как было показано в разд. 1, для базового элемента весьма важна функция переключения *Switch*, которая реализуется одновременно с функцией выбора *Select* одной из двух пар путей через базовый элемент. Аргументом этой функции являются значения бита $c_j \in \{0, 1\}$ запроса C_i, который определяет одну из двух пар путей через базовый элемент с переключением общей пары путей АФНФ либо без переключения.

Главный недостаток классического базового элемента заключается в неиспользовании им всех четырех случайных значений задержек ($\Delta(0)_{1,j}$, $\Delta(0)_{2,j}$, $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$), внесенных на этапе изготовления. В процессе функционирования АФНФ применяются только два, а именно $\delta_{0,j}$ и $\delta_{1,j}$ (2), из четырех приведенных ниже возможных случайных значений, представляющих собой добавленные значения задержки j-м базовым элементом:

$$\begin{aligned} &\delta(1)_j = \Delta(0)_{1,j} - \Delta(0)_{2,j},\ \delta(2)_j = \Delta(0)_{1,j} - \Delta(1)_{2,j}, \\ &\delta(3)_j = \Delta(1)_{1,j} - \Delta(0)_{2,j},\ \delta(4)_j = \Delta(1)_{1,j} - \Delta(1)_{2,j}. \end{aligned} \tag{3}$$

Отметим, что в данном случае рассматривается классический базовый элемент, для которого $\delta(1)_j = \delta_{0,j}$ и $\delta(4)_j = \delta_{1,j}$.

Как модификации базового элемента, состоящего из двух двухвходовых мультиплексоров, можно предложить структуры, изображенные на рис. 3. На первый взгляд, данные структуры повторяют друг друга, однако в их поведении, описываемом задержками (3), которые добавляет базовый элемент, наблюдаются заметные различия. Эти различия заключаются как в наборе функций базового элемента, так и в используемых им произвольных значениях $\Delta(0)_{1,j}$, $\Delta(0)_{2,j}$, $\Delta(1)_{1,j}$ и $\Delta(1)_{2,j}$ задержек, внесенных на стадии производства. Суммируя поведение классического базового элемента и четырех его модификаций (рис. 3), дадим их подробное описание (табл. 3).

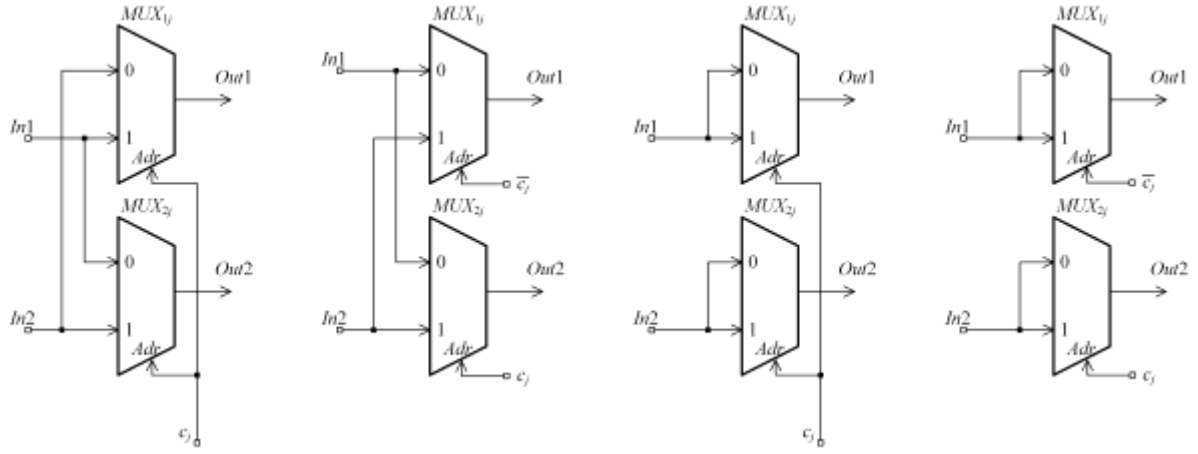

Рис. 3. Модификации базового элемента

Fig. 3. Base element modifications

Таблица 3
Описание базового элемента и его модификаций
Table 3
Description of basic element and its modifications

Базовый элемент *Basic Element*	Функция *Function*	Математическое описание *Mathematical description*	Значение $\delta_{c_j,j}$ *Value* $\delta_{c_j,j}$
Классический базовый элемент *Classical base element*	*Switch* *Select*	$d_j = d_{j-1} \cdot (-1)^{c_j} + \delta_{c_j,j}$	$\delta_{0,j} = \delta(1)_j = \Delta(0)_{1,j} - \Delta(0)_{2,j}$, $\delta_{1,j} = \delta(4)_j = \Delta(1)_{1,j} - \Delta(1)_{2,j}$
Модификация *a)* *Modification a)*	*Switch* *Select*	$d_j = d_{j-1} \cdot (-1)^{\overline{c_j}} + \delta_{c_j,j}$	$\delta_{0,j} = \delta(1)_j = \Delta(0)_{1,j} - \Delta(0)_{2,j}$, $\delta_{1,j} = \delta(4)_j = \Delta(1)_{1,j} - \Delta(1)_{2,j}$
Модификация *b)* *Modification b)*	*Switch* *Select*	$d_j = d_{j-1} \cdot (-1)^{\overline{c_j}} + \delta_{c_j,j}$	$\delta_{0,j} = \delta(3)_j = \Delta(1)_{1,j} - \Delta(0)_{2,j}$, $\delta_{1,j} = \delta(2)_j = \Delta(0)_{1,j} - \Delta(1)_{2,j}$
Модификация *c)* *Modification c)*	*Select*	$d_j = d_{j-1} + \delta_{c_j,j}$	$\delta_{0,j} = \delta(1)_j = \Delta(0)_{1,j} - \Delta(0)_{2,j}$, $\delta_{1,j} = \delta(4)_j = \Delta(1)_{1,j} - \Delta(1)_{2,j}$
Модификация *d)* *Modification d)*	*Select*	$d_j = d_{j-1} + \delta_{c_j,j}$	$\delta_{0,j} = \delta(3)_j = \Delta(1)_{1,j} - \Delta(0)_{2,j}$, $\delta_{1,j} = \delta(2)_j = \Delta(0)_{1,j} - \Delta(1)_{2,j}$

Возможны и другие модификации базового элемента, часть из которых неприменима для создания пары путей последовательным подключением базовых элементов, как это предполагает методология построения АФНФ. Некоторые модификации базового элемента могут быть получены несколькими путями. Так, модификация *a*) может быть получена при использовании классического элемента, у которого заменено обозначение входа *In*1 на *In*2 и наоборот, а соединение между элементами остается прежним. Следовательно, выходы предыдущего элемента *Out*1 и *Out*2 подключаются к входам *In*1 и *In*2 следующего модифицированного таким образом элемента.

Анализируя данные, приведенные в табл. 3, можно сделать очевидный вывод о том, что невозможно построить базовый элемент, который реализует обе функции *Switch* и *Select* и одновременно позволяет генерировать пару путей с одной из четырех добавленных задержек $\delta(1)_j$, $\delta(2)_j$, $\delta(3)_j$ или $\delta(4)_j$ (3). Это ограничение объясняется тем, что запрос c_j, подаваемый на *j*-й базовый элемент, имеет только два значения, которые позволяют выбрать один из двух, а не из четырех вариантов.

В качестве новых структур АФНФ можно рассматривать последовательное подключение различных базовых элементов и всевозможных их модификаций в различных сочетаниях. Таким образом нарушаются однородность и регулярность АФНФ, негативно сказывающиеся на их основных свойствах. Весьма интересны решения, которые используют базовые элементы, выполняющие обе функции *Switch* и *Select*, и базовые элементы, выполняющие только функцию *Select*.

Одним из многообещающих решений является модификация *d*), в которой мультиплексоры MUX_{1j} и MUX_{2j} управляются независимо, каждый своим битом c_{1j} и c_{2j} запроса $c_j = c_{1j}c_{2j}$. В этом случае запрос принимает четыре возможных значения, что позволяет использовать все четыре добавленные задержки $\delta(1)_j$, $\delta(2)_j$, $\delta(3)_j$ и $\delta(4)_j$ (3). Принимая во внимание значимость функции *Switch* как инструмента для нарушения структурной и топологической регулярности в АФНФ, ее аналог – операция *Switch* – реализуется путем соединений базовых элементов. Таким образом, выход *Out*1 *j*–1-й ступени подключается к входу *In*2 *j*-й ступени, а выход *Out*2 – к *In*1. Отметим, что операция *Switch* не повторяет функцию *Switch*, зависящую от запроса, но также является действенным инструментом для нарушения регулярности и повторяемости в схеме АФНФ. Особенно большое значение функция *Switch* имеет для реализаций АФНФ на FPGA. Функциональная схема подобной АФНФ, когда операция *Switch* используется на каждой ступени, изображена на рис. 4.

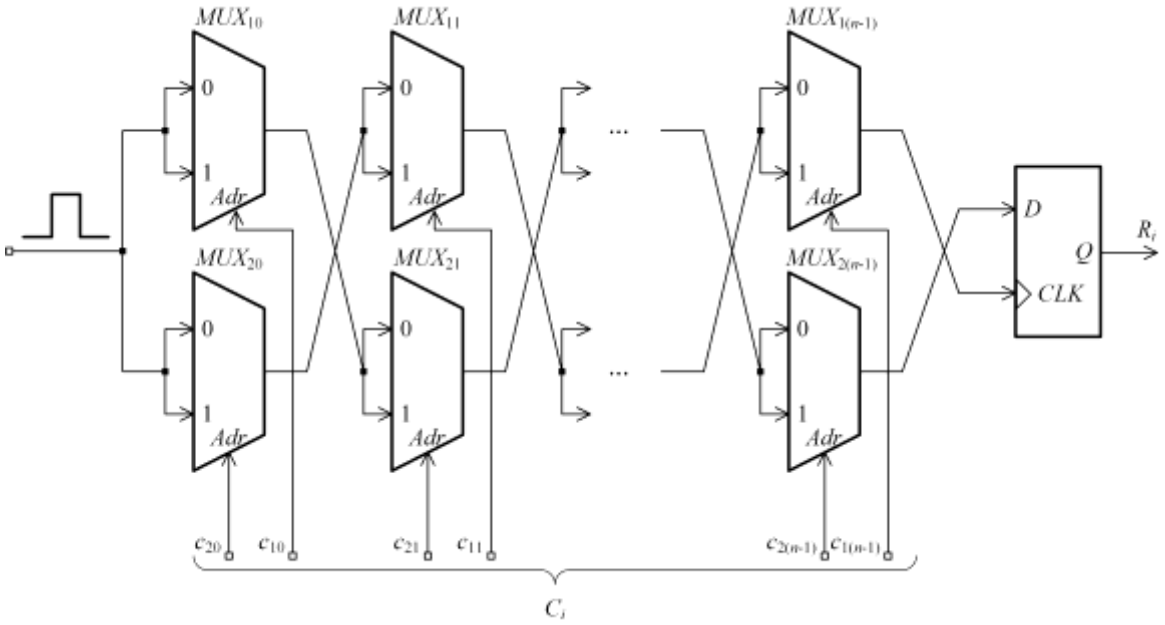

Рис. 4. Функциональная схема АФНФ

Fig. 4. Functional diagram of APUF

Математическая модель, описывающая АФНФ на рис. 4 рекурсивным соотношением для разницы задержки d_j прохождения сигнала по первому пути по отношению ко второму пути после j-й ступени, имеет вид

$$d_j = d_{j-1} \cdot (-1) + (\overline{c_{1j}}\,\overline{c_{2j}}) \cdot \delta(1)_j + (\overline{c_{1j}} c_{2j})\delta(2)_j + (c_{1j}\overline{c_{2j}}) \cdot \delta(3)_j + (c_{1j} c_{2j})\delta(4)_j, \quad (4)$$

где величины задержек представляют собой действительные числа, а переменные c_{1j} и c_{2j}, $j \in \{0, 1, 2, \ldots, n-1\}$, являются булевыми переменными, булевы операции над которыми формируют значения нуля либо единицы как одного из сомножителей.

Анализ изображенной на рис. 4 АФНФ показывает, что аппаратурная реализация подобной АФНФ характеризуется высокой эффективностью. На один бит n-разрядного запроса C_i необходим только один двухвходовый мультиплексор. Отметим, что все существующие решения для АФНФ требуют как минимум двух таких мультиплексоров [12−14].

Сопоставляя предложенную схему АФНФ с классической, отметим, что в первом случае базовый элемент j-й ступени всегда выполняет переключение путей, а во втором случае только при $c_j = 1$. В то же время в результате операции *Select* в предложенной структуре осуществляется выбор одного из четырех возможных добавленных значений задержки, а в классическом варианте – одного из двух.

Наличие двух битов c_{1j} и c_{2j} в запросе c_j для j-й ступени означает, что общее количество битов запроса C_i удваивается, а их общее число для АФНФ (рис. 4), состоящей из n последовательно подключенных базовых элементов, равняется 2^{2n}. Такую АФНФ можно рассматривать как двухмерную, так как конфигурация конкретной пары путей осуществляется и по горизонтали, и по вертикали.

3. Синтез базового элемента для 2D-АФНФ. В дальнейшем будем основываться на идее выбора одного из возможных значений задержки прохождения сигнала через многовходовый мультиплексор. Таким образом, базовый элемент будет включать многовходовые мультиплексоры с уникальными значениями задержки по их входам. Главным параметром базового элемента является количество реконфигурируемых путей прохождения сигнала через базовый элемент. Отметим, что не только две пары путей позволяют обеспечить эффективность АФНФ, как это имеет место в большинстве известных решений. Базовый элемент должен формировать множество независимых путей прохождения через него сигналов. Рассмотрение, например,

АФНФ с четырьмя одновременно функционирующими путями, очевидно, позволит получить новые положительные эффекты. Увеличение количества путей в АФНФ и извлечение из них ответов на запросы представляются перспективными направлениями дальнейших исследований.

Ключевым параметром базового элемента АФНФ является количество $Q \geq 2$ одновременно функционирующих путей при фиксированном запросе, подаваемом на него. Формально значение Q определяет число входов $In1, In2, \ldots, InQ$ базового элемента и такое же число его выходов $Out1, Out2, \ldots, OutQ$, а также количество мультиплексоров в базовом элементе и минимальное число входов в каждом из них. Функционально произвольный базовый элемент выполняет соединение входов с выходами, и эти соединения являются непересекающимися, а общее количество их разнообразных конфигураций определяется величиной $Q!$. Например, для $Q = 4$ одной из возможных конфигураций четырех путей прохождения сигнала через базовый элемент является *In*1-*Out*2, *In*2-*Out*3, *In*3-*Out*1 и *In*4-*Out*4, а максимальное количество подобных конфигураций 4! = 24. Для каждого адреса, одновременно подаваемого на входы четырех мультиплексоров, задается своя конфигурация путей базового элемента. Если базовый элемент строится, например, с использованием мультиплексоров с четырьмя входами, то конфигурация четырех путей строится для каждого из четырех адресов. Конфигурация *In*1-*Out*2, *In*2-*Out*3, *In*3-*Out*1 и *In*4-*Out*4 для значения адреса *Adr* четырехвходового мультиплексора, равного 00, означает, что вход *In*1 базового элемента подключен к нулевому (00) входу второго мультиплексора, *In*2 – к нулевому входу третьего мультиплексора, *In*3 – к одноименному входу первого мультиплексора и, наконец, *In*4 – к такому же входу четвертого мультиплексора. Графически четыре возможные конфигурации путей из 24 возможных для всех адресов базового элемента, построенного на четырех мультиплексорах с четырьмя входами каждый, показаны на рис. 5.

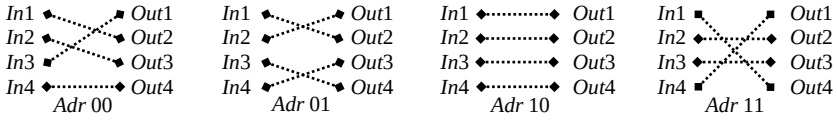

Рис. 5. Конфигурации путей базового элемента на четырех мультиплексорах

Fig. 5. Base element path configurations on four multiplexers

Конфигурации путей через базовый элемент, как отмечалось ранее, реализуются подключением входов $In1, In2, \ldots, InQ$ базового элемента к входам мультиплексоров, входящих в этот базовый элемент. Каждый из Q используемых мультиплексоров идентифицируется одним из выходов $Out1, Out2, \ldots, OutQ$ базового элемента, а входы – значениями его адресов *Adr*. Например, для рассмотренного выше примера каждый из четырех мультиплексоров имеет входы 00, 01, 10 и 11. Отметим что значение *Adr* в простейшем случае представляет собой запрос c_j, подаваемый на j-ю ступень АФНФ, которая представляет собой базовый элемент. Для рассмотренного выше примера базового элемента и его конфигураций путей (рис. 5) топология связей мультиплексоров с входами *In*1, *In*2, *In*3 и *In*4 базового элемента представлена в табл. 4.

Таблица 4

Топология связей мультиплексоров базового элемента

Table 4

Topology of multiplexer's connections of basic element

Вход *Input*	*Out*1	*Out*2	*Out*3	*Out*4
00	*In*3	*In*1	*In*2	*In*4
01	*In*2	*In*1	*In*4	*In*3
10	*In*1	*In*2	*In*3	*In*4
11	*In*4	*In*2	*In*3	*In*1

Анализ табл. 4 показывает, что при равенстве количества входов мультиплексора величине Q формально она должна удовлетворять требованию использовать в каждой строке таблицы все идентификаторы входов $In1, In2, \ldots, InQ$, причем каждый идентификатор – только один раз.

По умолчанию любой известный базовый элемент, применяемый для построения АФНФ, выполняет операцию *Select*, которая реализует выбор пары путей либо в общем случае большего количества путей базового элемента из множества возможных. В рассмотренном примере при фиксированном подключении мультиплексоров выбор осуществляется из четырех возможных конфигураций четырех путей (см. рис. 5). Отметим, что каждый из выбранных путей характеризуется своей индивидуальной задержкой сигнала, определяемой как вариациями параметров элементов, внесенных на этапе производства, так и их межсоединениями. Второй фактор имеет определяющее значение для реализации АФНФ на программируемых структурах [19]. Более того, возможность перепрограммирования подобных структур на FPGA раскрывает еще большие возможности для АФНФ.

На основе параметра Q, определяющего количество одновременно функционирующих путей, строится схема арбитра, формирующего ответы на подаваемые запросы. Для обеспечения высокой стабильности и непредсказуемости 2D-АФНФ в качестве эффективной схемы арбитра можно использовать, например, арбитр, основанный на операции сложения по модулю два (XOR arbiter) [26, 27]. Возможны и другие решения построения схемы арбитра в зависимости от количества генерируемых путей и формируемых битов ответа. Следует отметить, что увеличение количества битов ответа уменьшает стабильность и непредсказуемость 2D-АФНФ и, наоборот, его уменьшение повышает стабильность и одновременно непредсказуемость значений ответа.

Функция *Switch* в предлагаемых решениях базового элемента трансформируется в операцию реконфигурирования (*Reconfiguration*) путей, при которой возможны различные переключения путей из множества генерируемых.

Рассмотренная методика синтеза базового элемента основана на расширении функциональных возможностей АФНФ за счет увеличения альтернатив выбора возможных путей, т. е. не за счет количества последовательно подключенных базовых элементов, а за счет расширения возможностей выбора путей по запросу и за счет большего множества их вариантов. Для подтверждения данного тезиса в качестве простейшего примера рассмотрим классическую реализацию АФНФ, для которой построим альтернативную структуру 2D-АФНФ. Предположим, что в обоих случаях будут использоваться только два мультиплексора ($Q = 2$). Отличием 2D-АФНФ от классической ее реализации, приведенной на рис. 1, является применение для построения базового элемента вместо двухвходовых мультиплексоров с бóльшим количеством входов, например с четырьмя. Для этого случая базовый элемент будет реализовывать конкретную конфигурацию пар путей из множества возможных. Каждый адрес двух мультиплексоров с четырьмя входами базового элемента идентифицирует свою пару путей (рис. 6).

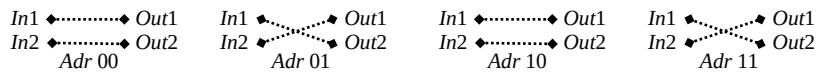

Рис. 6. Конфигурации путей базового элемента на двух мультиплексорах

Fig. 6. Base element path configurations on two multiplexers

В силу ограниченности количества путей (только два), проходящих через базовый элемент, число возможных конфигураций, формируемых им, также ограничено. В рассматриваемом примере их число равняется $2^4 = 16$, одна из указанных конфигураций изображена на рис. 6. Это означает, что при реализации 2D-АФНФ для анализируемого случая базового элемента его структура может быть разной, не повторяющейся для различных ее ступеней. На рис. 7 показана структура, соответствующая конфигурации путей на рис. 6.

Использование для случая $Q = 2$ мультиплексоров с восемью входами увеличивает число возможных конфигураций пар путей базового элемента до $2^8 = 256$. Во всех рассмотренных примерах базовый элемент реализует большее число уникальных пар путей. Следует отметить, что в качестве базовых элементов 2D-АФНФ могут быть предложены их реализации не только на мультиплексорах и не только с их общим количеством, равным размерности запроса C. Например, при реализации 2D-АФНФ эффективным представляется применение базовых элементов, использующих элементы И, ИЛИ и сумматор по модулю два [18].

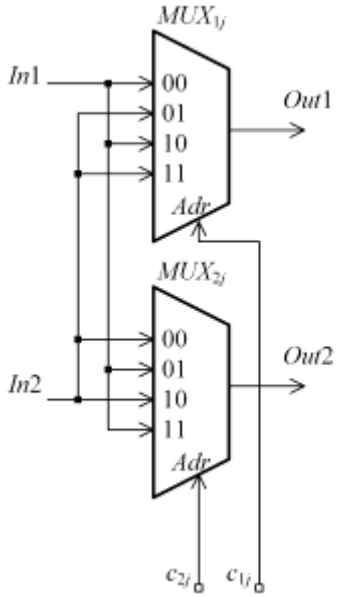

Рис. 7. Функциональная схема базового элемента
Fig. 7. Functional diagram of base element

Таким образом, для реальных структур 2D-АФНФ каждый их базовый элемент может быть реализован по уникальной, отличающейся от других базовых элементов, функциональной схеме и может управляться более чем одним битом запроса. В предельном случае 2D-АФНФ может быть реализована на одном базовом элементе, который для каждого из 2^n запросов C будет формировать свою уникальную пару путей.

4. Описание экспериментальных исследований. Для подтверждения предложенных в статье новых решений по построению 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. Общая структура аппаратуры экспериментальной установки изображена на рис. 8.

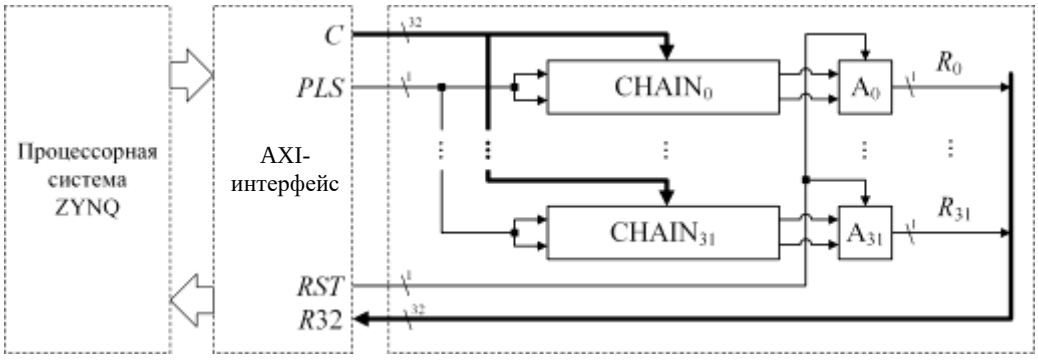

Рис. 8. Структурная схема экспериментальной установки
Fig. 8. Block diagram of experimental installation

Основу экспериментальной схемы составляют 32 реализации АФНФ ($APUF_j$), структурно разделенные на рисунке блоками $CHAIN_j$ и A_j, $j \in \{0,1,2,\ldots,31\}$. Каждый блок $CHAIN_j$ представляет собой n = 32 последовательно соединенные ступени схемы $APUF_j$, на которые подается 32-разрядное значение запроса C_i и фронт тестового импульса *PLS*. Выводы последней ступени каждого экземпляра $APUF_j$ подключены к независимым схемам арбитров A_j, которые, в свою очередь, вырабатывают значения ответов $R^i_j = APUF_j(C_i)$, объединенных в единую 32-разрядную шину ответов *R*32. Для инициализации схем арбитров применяется асинхронный сигнал *RST*, значение которого, как и значения сигнала *PLS* и шины *C*, формируется программно процессорной системой ZYNQ, входящей в состав кристалла FPGA.

Суть экспериментов заключалась в оценке такого важного показателя для схемных реализаций ФНФ, как стабильность [28]. Для его вычисления введем частоту встречаемости единичного ответа R^i_j = 1 при многократной подаче запроса C_i в E повторяющихся экспериментах:

$$P_1^E(C_i, j) = \frac{1}{E}\sum_{e=1}^{E} R_j^i \,. \tag{5}$$

Введем параметр нестабильности пары «запрос-ответ» $S^X(C_i, R_j^i) \in \{0,1\}$. Тогда если $0 < P_1^E(C_i, j) < 1$, то пара «запрос-ответ» (C_i, R^i_j) считается нестабильной и $S^X(C_i, R_j^i) = 1$. В противных случаях, когда $P_1^E(C_i, j) = 0$ либо $P_1^E(C_i, j) = 1$, пара считается стабильной и $S^X(C_i, R_j^i) = 0$.

Метрику стабильности конкретного схемного экземпляра $APUF_j$ будем вычислять следующим образом:

$$S(APUF_j) = 1 - \frac{1}{M}\sum_{m=1}^{M} S^X(C_m, R_j^m)\,, \tag{6}$$

где j = {0,1,2,…,31} – индекс АФНФ, M – число поданных уникальных запросов.

В ходе проведенных экспериментов были выбраны следующие параметры: число повторений каждого запроса E = 100 и число сгенерированных запросов $M = 10^5$. Все запросы были сгенерированы программной моделью 32-разрядного генератора псевдослучайной М-последовательности, которая обеспечивает уникальность и равномерность подаваемых запросов из 2^{32} возможных. В качестве схем арбитров A_j был использован синхронный *D*-триггер (технологический примитив FDCE). Оценке стабильности были подвергнуты три различные реализации блоков $CHAIN_j$: SCH_1, SCH_2 и SCH_3, которые были построены по схемам на рис. 1, 4 и 7 соответственно.

На рис. 9 изображены графики отсортированных по убыванию значений $S(APUF_j)$ для перечисленных схемных реализаций. Индексы схем АФНФ на оси абсцисс являются условными в силу осуществленной сортировки значений для трех различных реализаций. Для классической реализации АФНФ (схема SCH_1) только пять экземпляров из 32 обладают максимально возможным значением стабильности $S(APUF_j)$=1. Остальные 27 экземпляров имеют значения метрики стабильности, принадлежащие диапазону [0,99348; 0,99477]. Схемный вариант SCH_2 показал наличие 27 стабильных реализаций, а вариант SCH_3 – 16 стабильных реализаций из 32 возможных.

Данные, полученные в ходе проведенных экспериментов, показывают бóльшую стабильность в сравнении с классической реализацией АФНФ и состоятельность применения предложенных схемотехнических решений по построению 2D-АФНФ.

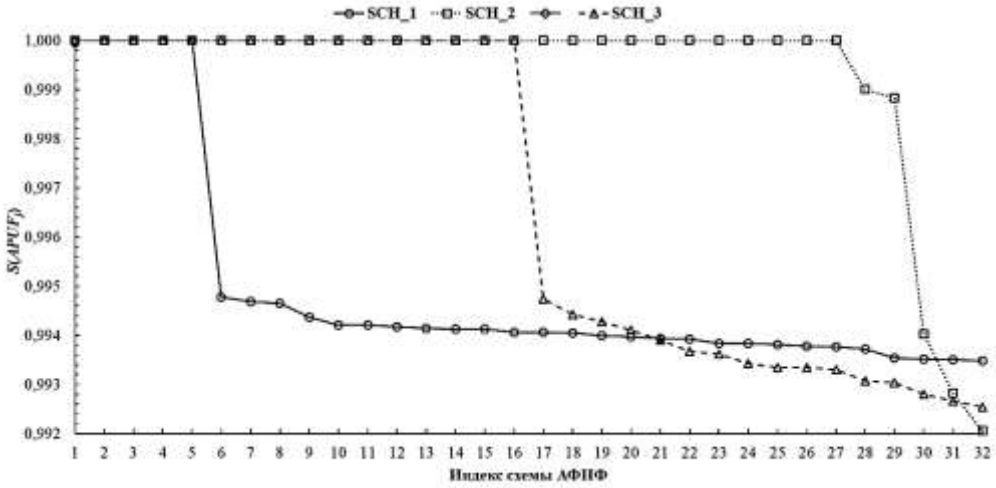

Рис. 9. Значения метрики стабильности $S(APUF_j)$ для различных схемных реализаций блоков $CHAIN_j$

Fig. 9. $S(APUF_j)$ stability parameter values for different implementation of $CHAIN_j$ blocks

Кроме параметра стабильности одной из важнейших характеристик ФНФ является соотношение единичных и нулевых ответов для различных подаваемых запросов. Подобная характеристика оценивается соответствующей метрикой единообразия (uniformity) [28]:

$$Un(APUF_j) = 1 - 2 \cdot \left| \frac{WH(R_j^M)}{M} - 0{,}5 \right|, \tag{7}$$

где $R_j^M = (R_j^0, R_j^1, R_j^2, ..., R_j^{M-1})$ – вектор ответов экземпляра ФНФ $APUF_j$ на однократно поданные M уникальных запросов, WH – вес по Хэммингу. Значение $Un(APUF_j) = 1$ достигается только при условии равенства числа полученных единичных и нулевых ответов соответствующего экземпляра $APUF_j$. Нулевое значение метрики (7) свидетельствует о равенстве всех ответов на поданные запросы.

На рис. 10 изображены графики отсортированных по убыванию значений $Un(APUF_j)$ для перечисленных схемных реализаций для $M = 10^5$ запросов. Индексы схем АФНФ на оси абсцисс также являются условными.

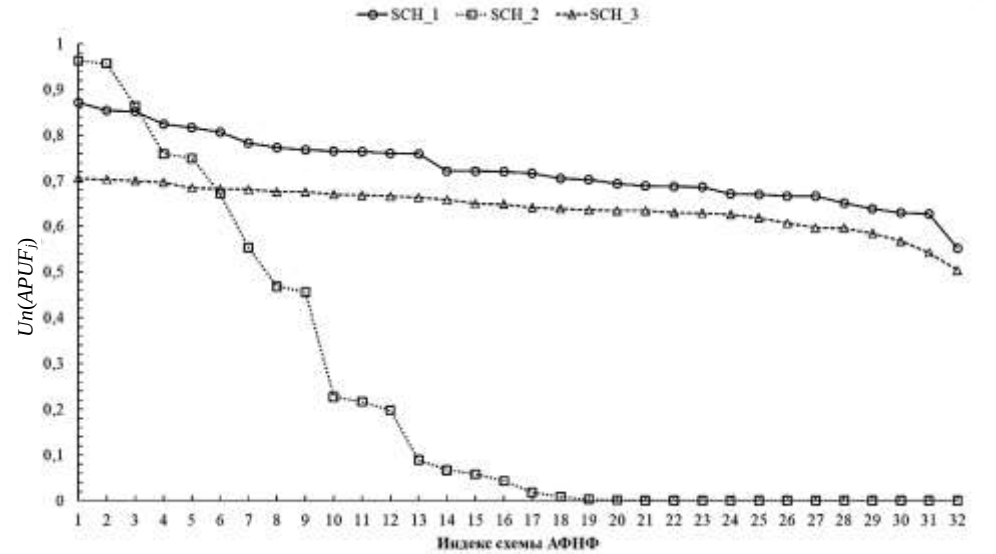

Рис. 10. Значения метрики единообразия $Un(APUF_j)$ для различных схемных реализаций блоков $CHAIN_j$

Fig. 10. $Un(APUF_j)$ uniformity parameter values for different implementation of $CHAIN_j$ blocks

На рис. 10 видно, что величины метрики $Un(APUF_j)$ сравнимы со значениями для классической АФНФ, а в отдельных случаях даже их превосходят.

Предложенные в статье схемы двухмерных АФНФ нуждаются в более детальном исследовании других статистических и вероятностных характеристик, в том числе при их реализациях на различных типах FPGA.

Заключение. В статье предложен подход к построению АФНФ, основанный на применении разнообразных модификаций базовых элементов. Главной характеристикой новых АФНФ является их двухмерность, заключающаяся не только в линейном наращивании количества базовых элементов, но и в придании более широких функциональных возможностей каждому базовому элементу. Интересным представляется дальнейшее развитие идеи построения 2D-АФНФ за счет комбинированного применения различных базовых элементов и реализации их на программируемых структурах типа FPGA.

Вклад авторов. *Ярмолик В. Н.* предложил идею построения двухмерных физически неклонируемых функций, *Иванюк А. А.* принял участие в обобщении и анализе полученных результатов и проведении экспериментальных исследований.

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 9th Computer and Communications Security Conf. (CCS'02), Washington, DC USA, 18–22 Nov. 2002. – Washington, 2002. – P. 148–160.

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

4. 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.

5. Skoric, B. 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. 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.

7. 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.

8. Maes, R. 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.

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

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

11. Программная реализация физически неклонируемых функций / Г. А. Мартвель [и др.] // Труды МФТИ. – 2020. – Т. 12, № 2. – С. 55–63.

12. Rührmair, U. On the foundations of Physical Unclonable Functions / U. Rührmair, J. Sölter, F. Sehnke // IACR Cryptology ePrint Archive. – 2009. – Vol. 2009. – 20 p.

13. Delvaux, J. 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.

14. 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.

15. Xu, X. 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.

16. Agarwal, A. Statistical timing analysis for intra-die process variations with spatial correlations / A. Agarwal, D. Blaauw, V. Zolotov // Proc. of Intern. Conf. on Computer Aided Design (ICCAD03), San Jose, CA, USA, 9–13 Nov. 2003. – San Jose, 2003. – P. 900–907.

17. 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.

18. 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.

19. Morozov, S. 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.

20. FPGA implementation of a cryptographically-secure PUF based on learning parity with noise / C. Jin [et al.] // Cryptography. – 2017. – Vol. 23, no. 1. – P. 1–20.

21. Gu, C. Improved reliability of FPGA-based PUF identification generator design / C. Gu, N. Hanley, M. O'neil // ACM Transactions on Reconfigurable Technology and Systems. – 2017. – Vol. 10, no. 3. – P. 1–23.

22. Kumar, A. METAPUF a challenge response pair generator / A. Kumar, S. L. Tripathi, R. Mishra // Periodicals of Engineering and Natural Sciences (PEN) . – 2018. – Vol. 2, no. 6. – P. 58–63.

23. Ozturk, E. Physical unclonable function with tristate buffers / E. Ozturk, G. Hammouri, B. Sunar // Proc. of IEEE Intern. Symp. on Circuits and Systems (ISCAS 2008), Seattle, Washington, USA, 18–21 May 2008. – Seattle, 2008. – P. 3194–3197.

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

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

26. A new Arbiter PUF for enhancing unpredictability on FPGA / T. Machida [et al.] // The Scientific World Journal. – 2015. – Vol. 2015, art. ID 864812. – 13 p.

27. Zhou, C. 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 54th ACM/EDAC/IEEE Design Automation Conf. (DAC 2017), Austin, TX, USA, 18–22 June 2017. – Austin, 2017. – P. 1–6.

28. Maiti, A. A systematic method to evaluate and compare the performance of Physical Unclonable Functions / A. Maiti, V. Gunreddy, P. Schaumont ; eds.: P. Athanas, D. Pnevmatikatos, N. Sklavos // Embedded Systems Design with FPGAs. – N. Y., Springer, 2013. – P. 245–267.

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., Kevenaar T. *Security with Noisy Data: On Private Biometrics, Secure Key Storage and Anti-Counterfeiting*. In P. Tuyls (ed.). New York, Springer, 2007, 339 p.

4. 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.

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. Lee J. W., Lim D., Gassend B., Suh G. E., …, 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.

7. Lim D., Lee J. W., Gassend B., Suh G. E., …, 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.

8. 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.

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

10. 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.).

11. Martvel G. A., Chuprakov F. M., Nedostoev K. A., Baburin N. S. *Software implementation of physically non-cloneable functions*. Trudy Moskovskogo fiziko-tehnicheskogo instituta [*Proceedings of Moscow Institute of Physics and Technology*], 2020, vol. 12, no. 2, pp. 55–63 (In Russ.).

12. Rührmair U., Sölter J., Sehnke F. On the foundations of Physical Unclonable Functions. *IACR Cryptology ePrint Archive*, 2009, vol. 2009, 20 p.

13. 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, 2013, pp. 137–142.

14. 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.

15. 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.

16. Agarwal A., Blaauw D., Zolotov V. Statistical timing analysis for intra-die process variations with spatial correlations. *Proceedings of International Conference on Computer Aided Design (ICCAD03), San Jose, CA, USA, 9–13 November 2003*. San Jose, 2003, pp. 900–907.

17. 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.).

18. 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.).

19. 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.

20. Jin C., Herder C., Ren L., Nguyen P. H., Fuller B., …, Dijk M. van. FPGA implementation of a cryptographically-secure PUF based on learning parity with noise. *Cryptography*, 2017, vol. 23, no. 1, pp. 1–20.

21. Gu C., Hanley N., O'neil M. Improved reliability of FPGA-based PUF identification generator design. *ACM Transactions on Reconfigurable Technology and Systems*, 2017, vol. 10, no. 3, pp. 1–23.

22. Kumar A., Tripathi S. L., Mishra R. METAPUF a challenge response pair generator. *Periodicals of Engineering and Natural Sciences (PEN)*, 2018, vol. 2, no. 6, pp. 58–63.

23. Ozturk E., Hammouri G., Sunar B. Physical unclonable function with tristate buffers. *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS 2008), Seattle, Washington, USA, 18–21 May 2008*. Seattle, 2008, pp. 3194–3197.

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

25. 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, vol. 20, no. 4, pp. 71–79 (In Russ.).

26. Machida T., Yamamoto D., Iwamoto M., Sakiyama K. A new Arbiter PUF for enhancing unpredictability on FPGA. *The Scientific World Journal*, 2015, vol. 2015, art. ID 864812, 13 p.

27. 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 54th ACM/EDAC/IEEE Design Automation Conference (DAC 2017), Austin, TX, USA, 18–22 June 2017.* Austin, 2017, pp. 1–6.

28. Maiti A., Gunreddy V., Schaumont P. A systematic method to evaluate and compare the performance of Physical Unclonable Functions. In P. Athanas, D. Pnevmatikatos, N. Sklavos (eds.). *Embedded Systems Design with FPGAs*. New York, Springer, 2013, pp. 245–267.

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, заведующий совместной учебной лабораторией «СК хайникс мемори солюшенс Восточная Европа», 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, Head of the Joint Educational Laboratory "SK Hynix Memory Solutions Eastern Europe", Belarusian State University of Informatics and Radioelectronics.
E-mail: ivaniuk@bsuir.by