УДК: 621.38

Дмитрий А. Козин[1], Павел В. Некрасов[2], Илья А. Митяков[3], Никита С. Пинчук[4]
Акционерное общество «Экспериментальное научно-производственное объединение СПЕЦИАЛИЗИРОВАННЫЕ ЭЛЕКТРОННЫЕ СИСТЕМЫ»,
Каширское ш., 31, Москва, 115409, Россия
[1]e-mail: dakoz@spels.ru, https://orcid.org/0009-0002-2856-9357
[2]e-mail: pvnek@spels.ru, https://orcid.org/0009-0004-3525-1077
[3]e-mail: mityakov_ia@mail.ru, https://orcid.org/0009-0004-7631-0117
[4]e-mail: pinchuk.nik@yandex.ru, https://orcid.org/0009-0007-0786-277X

ИССЛЕДОВАНИЕ ПАРАМЕТРОВ ФИЗИЧЕСКИ НЕКЛОНИРУЕМОЙ ФУНКЦИИ ТИПА АРБИТР ПРИ РЕАЛИЗАЦИИ НА ПЛИС

DOI: http://dx.doi.org/10.26583/bit.2023.2.10

Аннотация. Исследуется влияние различных схемотехнических решений физически не клонируемой функции (ФНФ) типа арбитр и расположение их логических блоков на параметры ФНФ. В связи с распространением и применением технологий интернета вещей (IoT) и соответственно устройств, подключённых к сотовой и интернет сети, актуальной проблемой является обеспечение безопасности и конфиденциальности доступа к данным. Следует отметить, устройства IoT требовательны к ряду параметров, таких как энергопотребление, геометрические размеры, необходимость защиты от кибератак. В качестве лёгкого, экономичного и повсеместного решения наиболее предпочтительными оказались методы аппаратной физической криптографии. Физически не клонируемые функции могут быть реализованы в различных устройствах путём проектирования их схемы на базе программируемых логических интегральных схем (ПЛИС). В статье использованы ПЛИС фирмы Xilinx семейства Zynq-7000. Данный подход даёт большую гибкость в отношении используемой архитектуры и позволяет более точно адаптировать систему к заданным требованиям. В качестве контрольно-измерительного оборудования использовались отладочные платы ПЛИС – EBAZ4205 и AntMiner_v1.0 на базе платформы XC7Z010CLG400-1 с двумя встроенными процессорными ядрами ARM Cortex-A9. Рассмотрен метод физической криптографии устройств IoT. Разработаны и реализованы конфигурационные прошивки ФНФ-арбитр с различным расположением функциональных блоков на кристалле. Проведена апробация конфигураций, показано изменение результатов в зависимости от расположения на кристалле. Ключевой задачей реализации ФНФ типа арбитр на любой аппаратной платформе является выравнивание длин линий распространения сигнала, которого можно добиться как добавлением дополнительных блоков задержки на линии, так и увеличением длины трассируемой линии сигнала. В данной работе наилучшие результаты получены при ручной трассировке и «вертикальном» расположении логических блоков схемы.

Ключевые слова: безопасность информации, физически неклонируемые функции (ФНФ), арбитр, программируемые логические интегральные схемы (ПЛИС), аппаратная криптография.
Для цитирования: КОЗИН, Дмитрий А.; НЕКРАСОВ, Павел В.; МИТЯКОВ, Илья А. ИССЛЕДОВАНИЕ ПАРАМЕТРОВ ФИЗИЧЕСКИ НЕКЛОНИРУЕМОЙ ФУНКЦИИ ТИПА АРБИТР ПРИ РЕАЛИЗАЦИИ НА ПЛИС. Безопасность информационных технологий, [S.l.], т. 30, № 2, с. 142–150, 2023. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/1509. DOI: http://dx.doi.org/10.26583/bit.2023.2.10.

Dmitry A. Kozin[1], Pavel V. Nekrasov[2], Ilya A. Mityakov[3], Nikita S. Pinchuk[4]
Joint Stock Company "Experimental Research and Production Association SPECIAL ELECTRONIC SYSTEM",
Kashirskoe sh., 31, Moscow, 115409, Russia
[1]e-mail: dakoz@spels.ru, https://orcid.org/0009-0002-2856-9357
[2]e-mail: pvnek@spels.ru, https://orcid.org/0009-0004-3525-1077
[3]e-mail: mityakov_ia@mail.ru, https://orcid.org/0009-0004-7631-0117
[4]e-mail: pinchuk.nik@yandex.ru, https://orcid.org/0009-0007-0786-277X

Investigation of the parameters of a physically unclonable arbiter-type function based on FPGA

DOI: http://dx.doi.org/10.26583/bit.2023.2.10

Abstract. The influence of various circuitry solutions of a physically unclonable function (PUF) of the arbiter type and the location of their logical blocks on the parameters of the PUF are investigated. The increasing proliferation and use of Internet of Things (IoT) technologies and, accordingly, devices connected to cellular and Internet networks, an urgent problem is to ensure the security and confidentiality of data access. It should be noted that IoT devices are demanding on a number of parameters, such as power consumption, integrated circuit geometric dimensions, and the requirement to protect against cyberattacks. As a simple, cost-effective and universal solution, hardware methods of physical cryptography proved to be the most preferred. Physically unclonable functions can be implemented in various devices by designing their circuits based on field-programmable gate arrays (FPGA). The Xilinx FPGA of the Zynq-7000 family has been used for this study. This approach gives greater flexibility in terms of the architecture used and allows the system to be more precisely adapted to any need. EBAZ4205 and AntMiner_v1.0 FPGA debug boards based on the XC7Z010CLG400-1 platform with two built-in ARM Cortex-A9 processor cores have been used as monitoring and measuring equipment. The method of physical cryptography of devices is considered. Several configuration bitstreams of PUF the arbiter-type firmware with different arrangement of functional blocks on a chip have been developed and implemented. All configurations have been tested, and the results were shown to change depending on the PUF location on the chip. An important task while implementing the PUF of the arbiter type on any hardware platform is to equalize the lengths of the signal propagation lines. This can be achieved both by adding additional delay blocks on the line, and by increasing the length of the routed signal line. In this study the best results were obtained with manual routing and a "vertical" arrangement of the logical blocks of the circuit.

Keywords: information security, internet of things (IoT), physically uncloneable functions (PUF), PUF-arbiter, FPGA, hardware cryptography.

For citation: KOZIN, Dmitry A.; NEKRASOV, Pavel V.; MITYAKOV, Ilya A. Investigation of the parameters of a physically unclonable arbiter-type function based on FPGA. IT Security (Russia), [S.l.], v. 30, no. 2, p. 142–150, 2023. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/1509. DOI: http://dx.doi.org/10.26583/bit.2023.2.10.

Введение

Концепция устройств интернета вещей (*Internet of Things*, IoT) подразумевает подключение их к сети для передачи данных между физическими объектами и центром управления (сервером), что позволяет автоматизировать множество процессов. По мере увеличения количества устройств подключённых к сети [1], актуальной проблемой становится обеспечение безопасности информации и конфиденциальности доступа к данным. В качестве лёгкого, экономичного и повсеместного решения наиболее предпочтительными оказались методы физической криптографии. Для аппаратной реализации данного метода используют физически неклонируемые функции (ФНФ) на базе ПЛИС.

1. Физически неклонируемые функции

Несмотря на относительную новизну данной концепции, в 2022 г. термин ФНФ отметил 20-летний юбилей, до сих пор отсутствует однозначное определение ФНФ. На практике одним из широко используемых является определение, согласно которому ФНФ представляют собой физические системы (устройства), обладающие неотъемлемым свойством неклонируемости некоторых их характеристик либо, чаще всего, параметров [2–3]. ФНФ выполняет функциональную операцию, то есть, запрашивая определённый ввод, выдает, соответствующий запросу, измеренный уникальный результат. Однако ФНФ не является математической функцией, это скорее некоторая процедура, которую

выполняет физическая система. Как правило, ввод в ФНФ называют вызовом/запросом, а вывод из функции – ответом, а соотношение, устанавливаемое между вызовами и ответами одной конкретной ФНФ – поведением пары вызов-ответ.

Свойства ФНФ

Каждая ФНФ обладает рядом важных свойств [3–4].

Уникальность

Уникальность ФНФ – вариация откликов или схемы ФНФ на один и тот же набор задач на разных микросхемах. В идеальном случае ответы двух экземпляров функций на один и тот же вызов должны отличаться в среднем на 50%, то есть между ними не должно существовать корреляции. В противном случае, если это свойство окажется слабым, то будет невозможно сгенерировать достаточное количество подписей для идентификации требуемого количества схем.

Неклонируемость

Неклонируемость – важное свойство ФНФ, указывающее на то, что создание двух идентичных схем, которые одинаково реагируют на одни и те же запросы, является очень сложной и практически нереализуемой задачей. Как правило ФНФ основано на различии микросхем ПЛИС из-за несовершенства производственного процесса и использует при вычислении отклика на воздействие параметры микросхемы (задержка распространения, емкость линий межсоединений, пороговые напряжения транзисторов и др.), которые варьируется от кристалла к кристаллу и являются неповторимыми даже внутри одной партии. Это обстоятельство указывает на то, что практически невозможно построить точную математическую модель ФНФ, которая будет предсказывать ответы на выбранные запросы без использования самой схемы функции, даже если даны точные параметры запроса и другие пары запросов-откликов.

Непредсказуемость

Согласно принципу непредсказуемости, ответ ФНФ на вызов должен быть непредсказуем, даже если известны изменения внешних воздействующих факторов, структура и расположение ФНФ. В дополнение к этому, система должна оставаться непредсказуемой, даже если известно бесконечно много пар вызов-ответ.

Стабильность

Количество битов в ответе, которые изменяют значение между многократно применяемыми вызовами. В идеальной ФНФ реакция на определенный запрос всегда должна быть одинаковой при любых условиях внешних факторов. Однако на практике ряд битов выставляется нестабильно, и они меняют свои значения в зависимости от температуры, напряжения питания, влажности, временной деградации и др. Это явление называется внутри кристальным шумом и требует применение постобработки данных.

Типы ФНФ

В настоящее время предложено более 40 типов ФНФ на различной основе (оптические, магнитные, кремниевые, на элементах памяти и др.). Чаще всего в цифровых устройствах используют ФНФ, основанные на внутренней неупорядоченности, происходящей в процессе производства.

Каждый из элементов можно охарактеризовать таким параметром, как задержка распространения сигнала. Одна из повсеместных вариаций построения ФНФ основывается на измерении задержек распространения сигнала в реконфигурируемых логических элементах устройства – цифровых путях [5–7].

ФНФ типа арбитр

В данном типе устанавливается условие гонки сигнала в электрической цепи, распространяющегося по двум различным цифровым путям. При этом арбитр обычно реализуют как триггер, который устанавливается в 0 или 1 в зависимости от того, какой переход завершится первым. Для реализации ФНФ на цифровом устройстве проектируют и собирают два топологически и функционально идентичных пути. Измерение времени распространения сигнала заключается в одновременной подаче на входы обоих путей одного и того же сигнала и определении, который из сигналов быстрее достигнет триггера [5, 8–9].

ФНФ на основе кольцевых генераторов

Значительно большей стабильностью характеризуются ФНФ на базе кольцевых генераторов. Функциональная схема данного типа ФНФ представляет собой нечётное число последовательно включенных инверторов, формирующих задержку, с положительной обратной связью, замыкающей кольцо [6–7]. Для определения начального состояния генератора на вход можно добавить логический элемент «И», один из входов которого является разрешающим, а второй используется для формирования кольца. Путём дублирования инверторов и добавления мультиплексоров между ними, можно сгенерировать экспоненциально большое количество пар запрос-ответ.

ФНФ на основе элементов памяти

Для генерации ключей и идентификаторов могут использоваться ФНФ, построенные на основе ячейки статического оперативно запоминающего устройства (СОЗУ) [10], просто собирая неинициализированные значения из цепей памяти. Например, каждый запоминающий элемент СОЗУ состоит из двух инверторов с перекрёстной обратной связью, выполненные в виде RS-триггера. После подачи питания ячейка памяти устанавливается в состояние 0 или 1. Причём в силу симметрии триггера это состояние невозможно предсказать, оно является случайным и определяется множеством факторов: изменением внутренних физических характеристик инверторов, неоднородностью кремниевой подложки, несовершенством технологического процесса изготовления транзисторов и т.п. Таким образом, подмножества блоков памяти служат вызовом, а считываемые с них значения – ответом. Однако, действительно случайными является только часть ячеек СОЗУ.

Производительность ФНФ

Ещё одним способом классификации ФНФ является производительность.

Обычно выделяют две категории ФНФ – сильные и слабые [11–13], классифицированные по количеству пар вызов-ответ, которую генерирует одно устройство. Это напрямую связано размером устройства и доступными им ресурсами.

Слабые ФНФ — это те устройства, которые поддерживают небольшое количество пар вызов-ответ. Кроме того, количество пар, которые генерирует слабая функция, масштабируются только линейно и при наличии физического доступа к устройству можно получить полный набор запросов. Часто слабые ФНФ используются для хранения ключей и аутентификации.

Сильные ФНФ — это устройства, которые могут поддерживать большое количество пар вызов-ответ. Количество пар может расти экспоненциально в зависимости от размера ФНФ. Важным отличием является то, что они могут поддерживать так много пар вызов-ответ, что, даже если бы злоумышленник имел доступ к устройству, он не смог бы записать их все. В сильных ФНФ каждую пару нужно использовать только один раз.

Очевидно, что получить идеальную ФНФ невозможно, так как при их реализации всегда нужно идти на компромисс между занимаемой площадью, энергопотреблением и стоимостью – а всё это влияет непосредственно на производительность.

2. Аппаратные платформы

Основной платформой для исследования и верификации различных схемотехнических решений ФНФ являются платы быстрого прототипирования цифровых устройств на основе кристаллов ПЛИС. Больший интерес представляют системы на кристалле (СнК), сочетающие в себе архитектуру ARM ядра (*Advanced RISC Machine)* и ПЛИС. Подобная комбинация позволяет строить системы, включающие как реализацию сложных алгоритмических задач на процессоре, так и обработку сигналов в реальном времени, для которых лучше всего подходит ПЛИС. Конечно, многие задачи можно решить, используя лишь достаточно мощный процессор с операционной системой реального времени или ПЛИС с подключенным программным IP-блоком процессорного ядра. Однако это требует дополнительных аппаратных затрат и высокий порог вхождения. Среди всех решений относительно недорогих микросхем можно выделить две аппаратно похожие системы семейства Cyclone V (производитель Altera) и Zynq-7000 (Xilinx). Более простым и доступным программным комплексом среди аналогичных решений других производителей является Xilinx, в связи с чем для дальнейшего использования было решено рассматривать системы на базе Zynq.

Выбор объекта исследования

При анализе доступных решений предпочтение было отдано отладочным платам начального уровня – EBAZ4205 и AntMiner_v1.0 построенные на платформе XC7Z010, и имеют следующие параметры и периферийные устройства:

- два физических ядра Cortex A9 666.66MHz;
- ПЛИС с 28 нм программируемой логикой,
- 28000 программируемых логических ячеек;
- 80 DSP блоков;
- DDR3 256 МБ внешней памяти MT41K128M16.

Интеграция устройств и обмен данными на ПЛИС осуществлялся с использованием физической процессорной системы, которой является Zynq-7000.

3. Исследования

Разработка конфигурационной прошивки

Для разработки конфигурационной прошивки были выбраны структуры ФНФ типа арбитр, ввиду простоты сборки, низкого энергопотребления, высокой скорости генерации ответа и большому числу выходных состояний по сравнению с остальными схемами [14–15].

Рассматривались классическая реализация ФНФ типа арбитр и 4-входовая ФНФ-арбитр. Для каждой из выбранных реализаций на языке описания аппаратуры Verilog было создано RTL-описание функции, состоящей из 16 пар последовательно подключенных мультиплексоров и D-триггера, в качестве арбитра. Результатом работы данной схемы (рис. 1) является один бит ответа ФНФ. С помощью данной функции сформированы пользовательскиеIP-блоки, с различной конфигурацией и подключением составных модулей, которые подключались к процессорной системе ПЛИС (рис. 2).

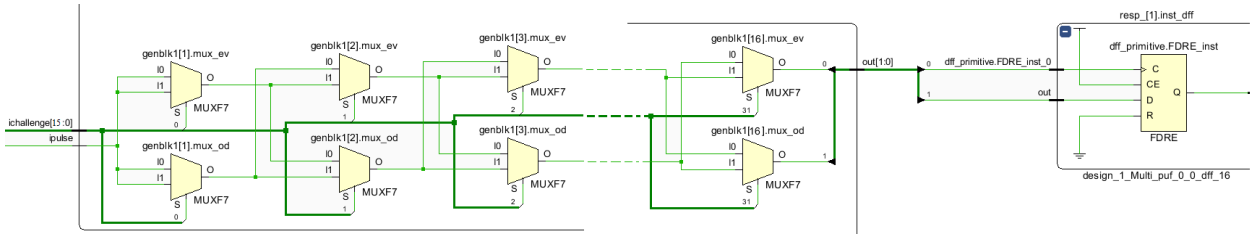


Рис. 1. Синтезированная схема ФНФ-арбитр
Fig. 1. Synthesized PUF-arbiter circuit

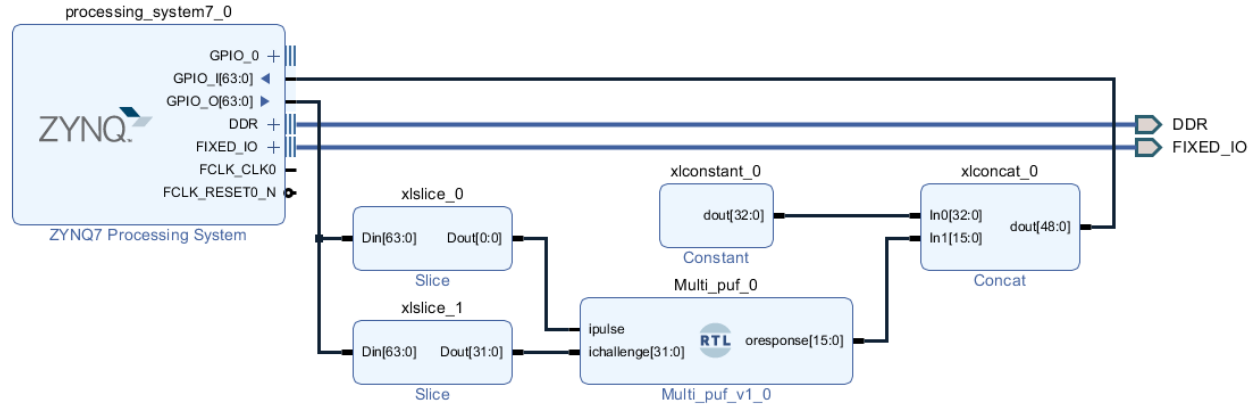


Рис. 2. Блок диаграмма проекта
Fig. 2. Block diagram of project

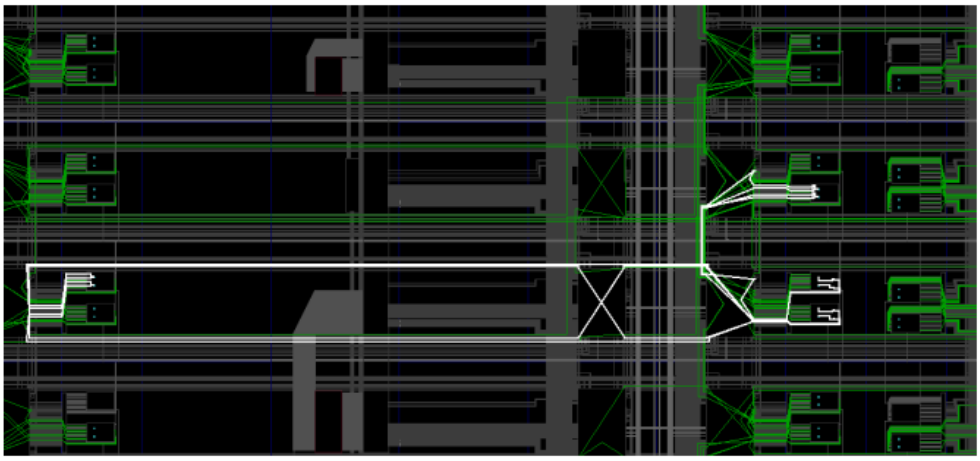
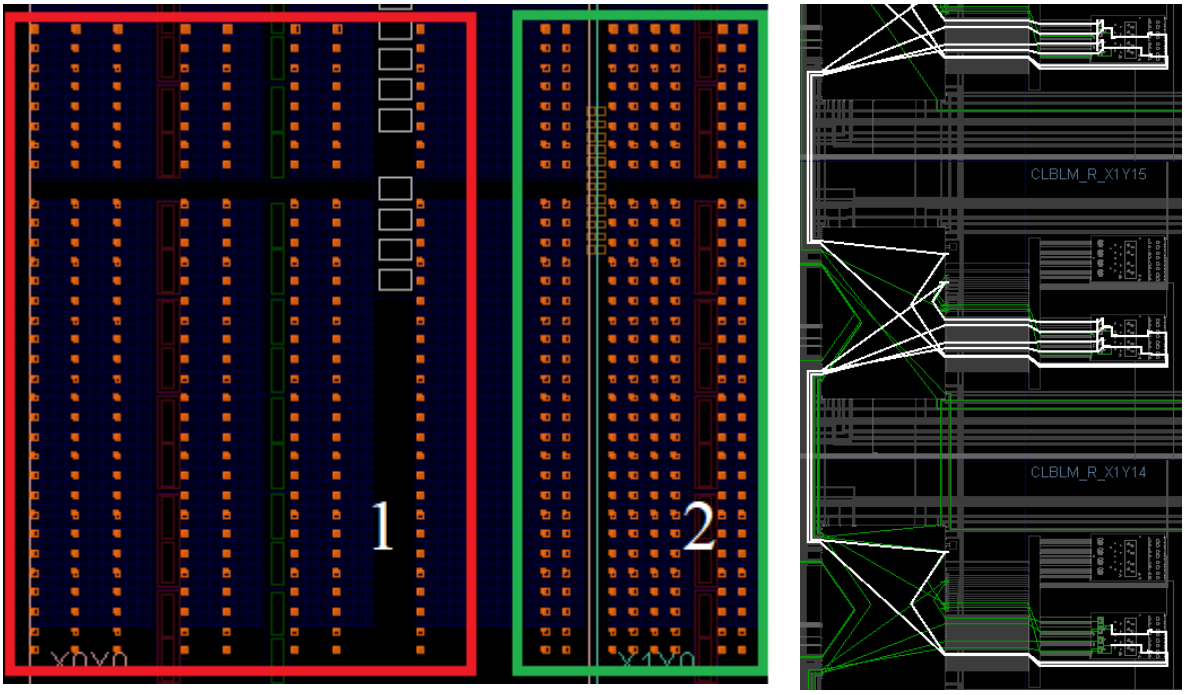


Рис. 3. Автоматическая трассировка логических блоков ФНФ
Fig. 3. Automatic tracing of PUF logical blocks

На рис. 3 представлено автоматическое расположение и маршрутизация логических блоков функции, использующая в качестве ресурсов – элементы MUXF7. Данная конструкция может быть значительно улучшена путём ручной настройки элементов арбитра (рис. 4), что гарантирует идентичную маршрутизацию каждого экземпляра функции. В таком случае задержки распространения сигнала действительно будут основаны на изменении физических процессов. Кроме улучшения характеристик, ручная настройка позволит упаковать арбитр в меньшее количество программируемых логических элементов, что уменьшит количество используемых ресурсов ПЛИС. Поэтому было реализовано несколько вариаций расположения функциональных блоков, среди которых

наилучший результат показала вертикальная трассировка «снизу-вверх». Далее будут рассмотрены именно эти конфигурации ФНФ.



(a) *(b)*

Рис. 4. (a) Макро-расположение логических блоков 1) Классическая ФНФ, 2) 4-входовая ФНФ. (b) Ручная трассировка логических блоков ФНФ
Fig. 4. (a) Macro-arrangement of logic blocks 1) Classical PUF, 2) 4-input PUF. (b) Manual tracing of PUF logical blocks

Апробация

Для проведения верификации работы различных конфигураций ФНФ, было разработано программное обеспечение. Об эффективности спроектированных функций можно судить по тестовым результатам в трёх основных категориях:

- Повторяемость (в идеальном случае должна быть приближена к 100%);
- Уникальность (в идеальном случае должна быть приближена к 50%);
- Стабильность (в идеальном случае должна быть приближена к 0%)

Эффективность ФНФ оценивалась с помощью подсчёта средних расстояний по Хеммингу, т.е. по количеству различающихся позиций в строках двух ответов одинаковой длины (табл. 1). Все измерения проводились в нормальных климатических условиях.

Таблица 1. Измеренные результаты

Параметры	Конфиг.1	Конфиг.2	Конфиг.3
Повторяемость(->100%)	75,6	98,4	93,8
Уникальность(->50%)	9,0	48,2	15,0
Стабильность(->0%)	2,3	0,3	0,8

В табл. 1 приведены: конфиг. 1 – классическая ФНФ-арбитр, с автоматической трассировкой; конфиг. 2 – классическая ФНФ-арбитр, с ручной трассировкой; конфиг. 3. – 4-входовой арбитр, с ручной трассировкой.

При увеличении количества пар до 32, значения параметров ФНФ меняется не более чем на два процента.

Было зафиксировано взаимовлияние блоков ФНФ, размещённых на одном кристалле, в результате чего ответы одного блока могли вносить искажения в работу другого блока. Это явление в значительной степени зависит от адресов конфигурационных блоков, используемых в схемах ФНФ, и требует дополнительных исследований.

Заключение

В ходе апробации было протестировано несколько конфигураций ФНФ типа арбитр реализованных на базе ПЛИС. Наилучших результатов работы ФНФ можно получить при ручной трассировке и «вертикальном» расположении логических блоков схемы. При построении ФНФ типа арбитр на любой аппаратной платформе ключевой задачей для достижения оптимальных характеристик является выравнивание длин линий распространения сигналов. Стабильность ответов ФНФ нелинейно зависит от аппаратных затрат и требует поиска оптимального значения. Так увеличение длины линии задержки в 2 раза с 16 до 32 пар мультиплексоров не дает существенных изменений стабильности ответов ФНФ и составляет не более 2%.

СПИСОКЛИТЕРАТУРЫ:

1. Hasan M. State of the IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally. IoTAnalytics, 2022. URL: https://iot-analytics.com/number-connected-iot-devices (дата обращения: 15.02.2023).
2. JinChenglu et al. FPGA Implementation of a Cryptographically-Secure PUF Based on Learning Parity with Noise. Cryptography 1,3 (2017 December): 23. URL:http://hdl.handle.net/1721.1/113338 (дата обращения: 15.02.2023).
3. Guajardo J., Kumar S.S., Schrijen GJ., Tuyls, P. FPGA Intrinsic PUFs and Their Use for IP Protection. In: Paillier, P., Verbauwhede, I. (eds) Cryptographic Hardware and Embedded Systems - CHES 2007. CHES 2007. Lecture Notes in Computer Science, vol. 4727. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-540-74735-2_5.
4. Иванюк А.А., Заливако С.С. Физическая криптография и защита цифровых устройств. Доклады БГУИР. 2019;(2):50-58. URL: https://doklady.bsuir.by/jour/article/view/1067 (дата обращения: 15.02.2023).
5. Ярмолик В.Н., Иванюк А.А., Шинкевич Н.Н. Физически неклонируемые функции с управляемой задержкой распространения сигналов. Информатика. 2022; 19(1):32-49. DOI: https://doi.org/10.37661/1816-0301-2021-19-1-32-49.
6. Komurcu Giray, Ali Emre Pusane and Gunhan Dundar. A ring oscillator based puf implementation on FPGA.IU-Journal of Electrical & Electronics Engineering13 (2013): 1647–1652. URL: https://electricajournal.org/Content/files/sayilar/65/1647-1652.pdf (дата обращения: 15.02.2023).
7. Ayat, Mehdi & Ebrahimi Atani, Reza & Mirzakuchaki, Sattar. (2011). On Design of PUF-Based Random Number Generators. International Journal of Network Security & Its Applications (IJNSA). 3. DOI: https://doi.org/10.5121/ijnsa.2011.3303.
8. Majzoobi M. and Koushanfar F. Time-Bounded Authentication of FPGAs. IEEE Transactions on Information Forensics and Security, vol. 6, no. 3, p. 1123–1135, 2011. DOI: https://doi.org/10.1109/TIFS.2011.2131133.
9. ИванюкА.А. Синтез симметричных путей физически неклонируемой функции типа арбитр на FPGA. Информатика. 2019, т. 16, № 2, с. 99–108. – EDN: LUJKXC.
10. Суханов С.В. Анализ физически неклонируемых функций на основе элементов памяти. Безопасность информационных технологий, [S.l.], т. 22, № 1, 2015. ISSN 2074-7136. URL: https://bit.mephi.ru/index.php/bit/article/view/130 (дата обращения: 15.02.2023). – EDN: VHHWEN.
11. Ярмолик В.Н., Вашинко Ю.Г. Физически неклонируемые функции. Информатика. 2011, № 2, с. 92–103. URL: https://inf.grid.by/jour/article/view/370 (дата обращения: 15.02.2023).
12. Заливако С.С., Иванюк А.А. Физически неклонируемые функции. Информационные технологии и системы 2019 (ИТС 2019) = Information Technologies and Systems 2019 (ITS 2019) : материалы международной научной конференции, Минск, 30 октября 2019 г. Белорусский государственный университет информатики и радиоэлектроники; редкол.: Л. Ю. Шилин [и др.]. Минск, 2019, с. 8–21. URL: https://libeldoc.bsuir.by/handle/123456789/37034 (дата обращения: 15.02.2023).
13. Бельский В.С., Чижов И.В., Чичаева А.А., Шишкин В.А. Физически неклонируемые функции в криптографии. International Journal of Open Information Technologies. 2020, № 10.

URL: https://cyberleninka.ru/article/n/fizicheski-nekloniruemye-funktsii-v-kriptografii (дата обращения: 15.02.2023).
14. Aknesil C. and Dubrova E. An FPGA Implementation of 4×4 Arbiter PUF. IEEE 51st International Symposium on Multiple-Valued Logic (ISMVL), Nur-sultan, Kazakhstan. 2021, p. 160–165. DOI: https://doi.org/10.1109/ISMVL51352.2021.00035.
15. Ярмолик Вячеслав Н.; Иванюк Александр А. Сбалансированные физически неклонируемые функции типа арбитр. Безопасность информационных технологий, [S.l.], т. 30, № 1, с. 92–107, 2023. ISSN 2074-7136. DOI: http://dx.doi.org/10.26583/bit.2023.1.07. – EDN: HCUPGE.

REFERENCES:

[1] Hasan M. State of the IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally. IoT Analytics, 2022. URL: https://iot-analytics.com/number-connected-iot-devices (accessed: 15.02.2023).
[2] JinChenglu et al. FPGA Implementation of a Cryptographically-Secure PUF Based on Learning Parity with Noise. Cryptography 1,3 (2017 December): 23. URL: http://hdl.handle.net/1721.1/113338 (accessed 15.02.2023).
[3] Guajardo J., Kumar S.S., Schrijen GJ., Tuyls P. FPGA Intrinsic PUFs and Their Use for IP Protection. In: Paillier, P., Verbauwhede, I. (eds) Cryptographic Hardware and Embedded Systems - CHES 2007. CHES 2007. Lecture Notes in Computer Science, vol 4727. Springer, Berlin, Heidelberg. DOI: https://doi.org/10.1007/978-3-540-74735-2_5.
[4] Ivaniuk A.A., Zalivaka S.S. Physical cryptography and security of digital devices. Doklady BGUIR. 2019;(2):50-58. URL: https://doklady.bsuir.by/jour/article/view/1067 (in Russian).
[5] Yarmolik V.N., Ivanyuk A.A., Shinkevich N.N. Physically non-cloneable functions with controlled propagation delay. Computer science. 2022;19(1):32-49. DOI: https://doi.org/10.37661/1816-0301-2021-19-1-32-49.
[6] Komurcu Giray, Ali Emre Pusane and Gunhan Dundar. A ring oscillator based puf implementation on FPGA. IU-JournalofElectrical&ElectronicsEngineering13(2013):1647–1652.
URL: https://electricajournal.org/Content/files/sayilar/65/1647-1652.pdf (accessed: 15.02.2023).
[7] Ayat Mehdi & Ebrahimi Atani Reza & Mirzakuchaki Sattar. (2011). On Design of PUF-Based Random Number Generators. International Journal of Network Security & Its Applications (IJNSA). 3. DOI: https://doi.org/10.5121/ijnsa.2011.3303.
[8] Majzoobi M. and Koushanfar F. Time-Bounded Authentication of FPGAs. IEEE Transactions on Information Forensics and Security, vol. 6, no. 3, p. 1123–1135, 2011. DOI: https://doi.org/10.1109/TIFS.2011.2131133.
[9] Ivaniuk A.A. Synthesis of symmetric paths of arbiter physically unclonable function on FPGAInformatics. 2019, v. 16, no. 2, p. 99–108 (in Russian). – EDN: LUJKXC.
[10] Sukhanov S.V. Analysis of physical unclonable functions based on memory. IT Security (Russia), [S.l.], v. 22, no. 1, p. 47–51, 2015. URL: https://bit.mephi.ru/index.php/bit/article/view/130 (accessed: 20.02.2023) (in Russian). – EDN: VHHWEN.
[11] Yarmolik V.N., Vashinko Yu.G. Physically non-cloneable functions. Informatics. 2011, no. 2, p. 92–103. URL: https://inf.grid.by/jour/article/view/370 (accessed 15.02.2023) (in Russian).
[12] Zalivako S.S., Ivanyuk A.A. Physically non-cloneable functions. Information Technologies and Systems 2019 (ITS 2019) = Information Technologies and Systems 2019 (ITS 2019): materials of the international scientific conference, Minsk, October 30, 2019. Belarusian State University of Informatics and Radioelectronics; redol.: L.Yu.Shilin[Idr.].Minsk, 2019, p. 8–21. URL: https://libeldoc.bsuir.by/handle/123456789/37034 (accessed 15.02.2023) (in Russian).
[13] Belsky V.S., Chizhov I.V., Chichaeva A.A., Shishkin V.A. Physically non-clonable functions in cryptography. International Journal of Open Information Technologies. 2020, no. 10. URL: https://cyberleninka.ru/article/n/fizicheski-nekloniruemye-funktsii-v-kriptografii (accessed: 20.02.2023) (in Russian).
[14] Aknesil C. and DubrovaE.An FPGA Implementation of 4×4 Arbiter PUF. IEEE 51st International Symposium on Multiple-Valued Logic (ISMVL), Nur-sultan, Kazakhstan. 2021, p. 160–165. DOI: https://doi.org/10.1109/ISMVL51352.2021.00035.
[15] Yarmolik Vyacheslav N.; Ivaniuk Alexander A. Balanced arbiter physical uncloneable functions. IT Security (Russia), [S.l.], v. 30, no. 1, p. 92–107, 2023. ISSN 2074-7136. DOI: http://dx.doi.org/10.26583/bit.2023.1.07 (in Russian). – EDN: HCUPGE.

Поступила в редакцию – 10 апреля 2023 г. Окончательный вариант – 04 мая 2023 г.
Received – April 10,2023. The final version – May 04, 2023.