ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫЕ ФУНКЦИИ

Заливако С. С., Иванюк А. А.
Кафедра информатики, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: {zalivako, ivaniuk}@bsuir.by

В статье рассмотрены базовые компоненты физической криптографии цифровых устройств – физически неклонируемые функции (ФНФ). Интерес к данной тематике обусловлен возможностью извлечения уникальных характеристик интегральных схем (ИС) для построения неклонируемых идентификаторов и генерирования случайных числовых последовательностей. Как правило, схемотехнические реализации ФНФ гораздо более эффективны с точки зрения аппаратурных затрат в сравнении с классическими алгоритмами шифрования и хеширования. Приведен обзор основных областей применения ФНФ, а также описаны методы экспериментального исследования архитектур ФНФ на FPGA (Field-Programmable Gate Array).

Введение

Количество устройств Интернета вещей (Internet of Things, IoT), подключенных к сети, в 2018 составило порядка 7 млрд [1]. По прогнозу компании Ericsson [2] их число к 2020 году увеличится до 30 млрд в первую очередь за счет развития мобильной связи пятого поколения. В связи с повсеместным распространением Интернета вещей одной из важнейших проблем в настоящее время является безопасность и конфиденциальность доступа к хранимым устройствами данным [3]. Решением этой проблемы является применение методов аппаратной и программной криптографии: шифрования, обфускации, внедрения цифровых водяных знаков и отпечатков пальцев, идентификации, аутентификации и др. [4]. Поскольку рассматриваемый класс устройств является требовательным к энергопотреблению и площади кристалла интегральной схемы (ИС), методы физической криптографии оказались более предпочтительными [5]. Для аппаратной реализации методов физической криптографии, как правило, используются физически неклонируемые функции (ФНФ), которые предназначены для генерирования секретных ключей и идентификаторов цифровых устройств для их последующей аутентификации [6].

В данной статье приведены результаты научных исследований по тематике физически неклонируемых функций под руководством профессора Иванюка А. А. в период с 2012 по 2019 годы. Впервые на постсоветском пространстве проблематика ФНФ была опубликована профессором Ярмоликом В. Н. в 2011 году в журнале “Информатика” [7]. Профессор Ярмолик является ведущим специалистом в области проектирования надежных цифровых устройств и систем.

I. Физически неклонируемые функции

Формально ФНФ может быть описана значениями пар входных и соответствующих им выходных параметров, которые для ФНФ, реализованных в полупроводниковых устройствах, являются соответственно значениями входных сигналов запроса CH (Challenge) и значениями выходных сигналов ответа R (Response) [8]. Любая ФНФ может быть представлена с помощью множества всевозможных пар запрос-ответ (Challenge-Response Pairs, CRP), а также является функцией преобразования множества запросов CH_i во множество ответов R_i:

$$R_i = \mathrm{PUF}(CH_i) \tag{1}$$

В силу того, что современные ИС обладают множеством физических характеристик, точные значения которых являются уникальными и непредсказуемыми для каждой произведенной копии, эти параметры могут быть использованы в качестве основы для реализации ФНФ. Исторически первыми были оптические ФНФ, основанные на направленной под определенным углом световой волне (запрос) для получения уникальной интерференционной картины (ответ) [9]. В работе [10] впервые было предложено использовать различия в задержке распространения сигнала по симметричным путям для реализации ФНФ типа арбитр (Arbiter PUF). Эта идея в дальнейшем была использована, например, в ФНФ, основанной на таблицах поиска (Lookup Table PUF) [11]. Разность частот кольцевых генераторов (Ring Oscillator PUF) [12], а также уникальность значений частот (Bistable Ring PUF) [13] были использованы в качестве основы для генерирования пар запрос-ответ. Множество реализаций ФНФ основано на использовании состояния элементов памяти после инициализации: ФНФ на основе статического оперативного запоминающего устройства (СОЗУ) (SRAM PUF) [14], динамической памяти с произвольным доступом (DRAM PUF) [15], ФНФ типа бабочки (Butterfly PUF) [16], ФНФ на основе асинхронного RS-триггера (SR latch PUF) [17], а также магнитно-резистивной оперативной памяти, основанной на спиновых эффектах (Spin-Transfer Torque Magnetoresistive Random-Access Memory) [18]. Также влияние качественного и количественного состава покрытия ИС на значения

электрической емкости ее элементов (Coating PUF) [19] было успешно использовано для реализации идентификации и аутентификации ИС с помощью ФНФ.

С учетом многообразия возможных реализаций ФНФ на кристалле ИС следует выделить несколько областей применения ФНФ: цифровые водяные знаки и отпечатки пальцев [20], генерирование случайных числовых последовательностей [21], идентификация и аутентификация [22], реализация аппаратных хэш-функций [23], обнаружение аппаратных закладок [24], генерирование ключей шифрования [25], радиочастотные идентификаторы [26] и др. Более того, ведущие мировые производители полупроводниковых устройств применяют ФНФ в устройствах интернета вещей (Samsung [27]), серийно выпускаемых программируемых логических интегральных схемах (Intel [28], Xilinx [29]), защите цифровых устройств от нелегального копирования (Qualcomm [30]), устройствах радиочастотной идентификации (Intrinsic ID [31]), цифровых узлах транспортных средств (Accenture [32]) и других разработках.

В данной статье рассматривается неклонируемая идентификация и аутентификация, схемотехнические решения, направленные на снижение уязвимости к криптографическим атакам, генерирование случайных числовых последовательностей, а также экспериментальные исследования ФНФ с помощью аппаратно-программных комплексов.

II. Неклонируемая идентификация цифровых устройств

Классическая реализация ФНФ типа арбитр (АФНФ) основана на сравнении задержек двух идентичных сигналов, распространяемых по топологически идентичным путям [10]. Схемная реализация АФНФ приведена на рис. 1.

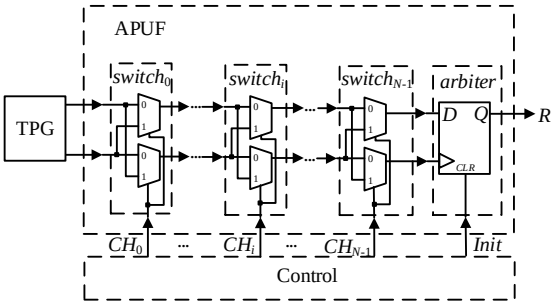

Рис. 1 – Классическая реализация ФНФ типа арбитр

Модуль TPG (Test Pulse Generator) предназначен для генерирования переднего и/или заднего фронта сигнала в зависимости от реализации схемы арбитра. Каждый из переключателей $switch_i$ имеет два режима работы: прямой — когда сигнал распространяется по тому же пути, и перекрестный – в этом случае сигнал меняет линию связи. В качестве элемента, определяющего, какой из сигналов пришел быстрее, используется синхронный D-триггер, который при небольших значениях (точные значения зависят от технологического процесса изготовления ИС) разницы задержек сигналов переходит в метастабильное состояние и тем самым значение на его выходе становится непредсказуемым. В результате данного эффекта показатель стабильности АФНФ, реализованных на ПЛИС, не превышает 0,6 при максимальном значении 1,0.

Для обнаружения метастабильного состояния арбитра было предложено два схемотехнических решения [33]. Одно из решений, основанное на применении четырех D-триггеров, приведено на рис. 2.

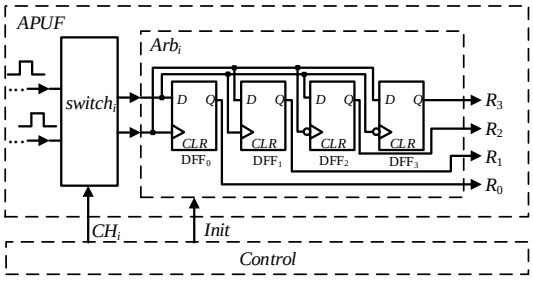

Рис. 2 – Реализация арбитра на основе четырех D-триггеров

Предлагаемая реализация арбитра стробируется не только передним (триггеры DFF_0 и DFF_1), но и задним фронтом (триггеры DFF_2 и DFF_3) тестового импульса. Следовательно, значения на выходе арбитра показывают взаимоотношение целых импульсов, а не только их части (передних фронтов), как в классической реализации АФНФ. Таким образом, ответ модифицированного арбитра может быть представлен четырехразрядным числом $\{R_0, R_1, R_2, R_3\}$. Предложенная реализация арбитра на базе четырех D-триггеров позволяет обнаруживать метастабильные состояния, которые, в свою очередь, могут быть обозначены как третий символ выходного алфавита (X) в дополнение к имеющимся 0 и 1.

Другим решением для обнаружения метастабильного состояния является реализация схемы арбитра в виде асинхронного RS-триггера, как показано на рис. 3. Предложенный арбитр состоит из двух логических элементов ИЛИ-НЕ с обратной связью, а также двух D-триггеров. В отличие от классической реализации АФНФ предлагаемая схема арбитра управляется задним фронтом тестового сигнала. Как показано в работе [34], метастабильное состояние на выходе RS-триггера наблюдается в случае, когда сигнал

на входах установки и сброса одновременно переходит из состояния логической единицы в состояние логического ноля. Признаком метастабильного состояния является затухающее колебание, производимое RS-триггером. Если явление метастабильности не было обнаружено, то RS-триггер мгновенно перейдет в состояние стабильного логического ноля или единицы.

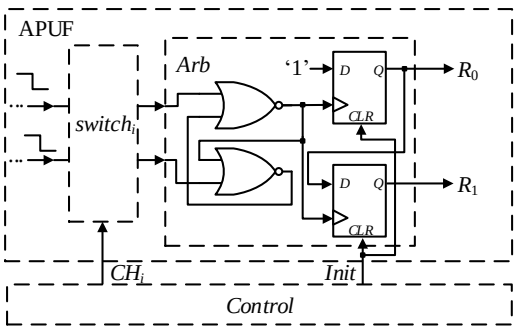

Рис. 3 – Реализация арбитра на основе асинхронного RS-триггера

Таким образом, на выходе арбитра возможно появление трех стабильных состояний: логического ноля, логической единицы и высокочастотного затухающего колебания (High Frequency Oscillation, HFO). Обнаружение перечисленных состояний может быть реализовано с помощью сдвигового регистра, состоящего из двух D-триггеров: стабильный ноль кодируется как пара ($R_0 = 0$, $R_1 = 0$), стабильная единица — как ($R_0 = 1$, $R_1 = 0$), а состояние HFO — как ($R_0 = 1$, $R_1 = 1$). Следовательно, значение бита ответа R_1 показывает, что арбитр находится в стабильном (когда $R_1 = 0$) или метастабильном ($R_1 = 1$) состоянии.

По результатам тестирования предложенная реализация АФНФ с модифицированным арбитром как на базе четырех D-триггеров, так и RS-триггера, продемонстрировала высокие значения характеристик уникальности (≈0,49) и стабильности (≈0,99). Таким образом, в результате характеристика стабильности была значительно улучшена с 0,57 до 0,99 с учетом дополнительных аппаратурных затрат, не превышающих 2 % от исходной реализации схемы арбитра.

Использование ФНФ в таких приложениях, как медицинская электроника [35], RFID-карты доступа к объектам повышенной секретности [36], системы управления стратегическими объектами (например, атомными электростанциями) [37] и т. п. требует предельно высокого уровня стабильности ее ответов (≈1,0).

Модифицированные схемы арбитров для обнаружения метастабильных состояний позволяют значительно улучшить характеристики стабильности, но, к сожалению, не обеспечивают требуемого уровня стабильности. В связи с этим появляется необходимость в разработке более точной модели АФНФ не только для обнаружения метастабильных состояний, но и их предсказания с вероятностью не менее 0,95 по виду подаваемого запроса. Уточненная математическая модель задержки времени распространения сигнала позволила разработать критерии для тестирования запросов с целью их классификации на менее стабильные (слабые) и более стабильные (сильные).

На рис. 4 показана структурная схема АФНФ.

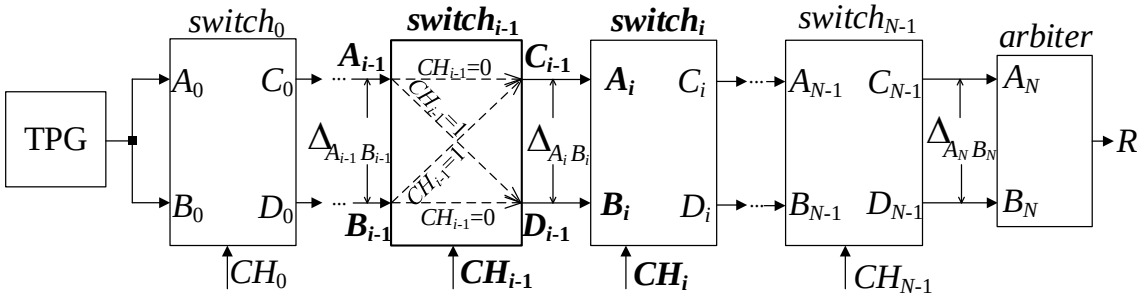

Рис. 4 – Структурная схема ФНФ типа арбитр

Согласно определению ФНФ, множества значений $\{\Delta_{C_{i-1}A_{i-1}}, \Delta_{D_{i-1}A_{i-1}}, \Delta_{C_{i-1}B_{i-1}}, \Delta_{D_{i-1}B_{i-1}}\}$ и $\{\Delta_{A_iC_{i-1}}, \Delta_{B_iD_{i-1}}\}$ $\forall i = 1, \ldots, N$ являются уникальными и неповторяющимися не только для схемной реализации всех звеньев одной схемы на одном полупроводниковом кристалле, но и на множестве кристаллов.

В зависимости от подаваемого значения CH_{i-1} формируются два уникальных маршрута прохождения двух тестовых импульсов от входных портов A_{i-1}, B_{i-1} до портов следующей компоненты A_i, B_i. Как было показано выше, на выходе арбитра R может появиться метастабильное состояние, что ухудшает характеристику стабильности ФНФ. Такая ситуация происходит в

результате того, что разность во времени задержки сигналов, распространяющихся по симметричным путям, попадает в интервал $[t_{low}, t_{high}]$. В свою очередь, значения t_{low} и t_{high} определяются временными характеристиками схемы арбитра. Для классической реализации с помощью единственного D-триггера значение $t_{low} = -t_{hold}$, а значение $t_{high} = t_{setup}$ (t_{setup}, t_{hold} – время предустановки и удержания триггера, соответственно). Если же арбитр был реализован на основе RS-триггера, то параметры t_{low} и t_{high} определяются в зависимости от характеристик напряжения и симметричности реализации логических элементов ИЛИ-НЕ, которые лежат в основе реализации RS-триггера [34].

Учет перечисленных параметров для схемы арбитра гарантирует появление стабильного значения ответа на выходе R. В противном случае схема арбитра может оказаться в метастабильном состоянии, при котором значение ответа на выходе R будет непредсказуемым. В итоге значение ответа на выходе R зависит от результирующей разницы между фронтами сигналов $\Delta_{A_N B_N}$:

$$R = \begin{cases} 0, & \text{если } \Delta_{A_N B_N} \leq t_{low}, \\ 1, & \text{если } \Delta_{A_N B_N} \geq t_{high}, \\ \text{X}, & \text{если } t_{low} < \Delta_{A_N B_N} < t_{high}. \end{cases} \quad (2)$$

Значение результирующей разницы $\Delta_{A_i B_i}$ для блока $switch_i$ формально можно выразить следующей функцией γ от двух аргументов:

$$\Delta_{A_i B_i} = \gamma(\delta_{i-1}^{CH_{i-1}}, \Delta_{A_{i-1} B_{i-1}}), \quad (3)$$

где $\delta_{i-1}^{CH_{i-1}}$ – уникальная характеристика звена $switch_{i-1}$, значение которой зависит от бита запроса CH_{i-1}; $\Delta_{A_{i-1} B_{i-1}}$ – временная разница фронтов сигналов на входе звена $switch_{i-1}$.

В свою очередь, значение $\Delta_{A_N B_N}$ можно выразить как:

$$\Delta_{A_N B_N}(CH_{N-1}, CH_{N-2}, \ldots, CH_0) = \\ = \sum_{j=0}^{N-1} (\delta_j \prod_{i=0}^{j} Sign_i). \quad (4)$$

Функция арифметического знака задержки $Sign_i$ может быть представлена как:

$$Sign_{i-1} = 1 - 2 \cdot CH_{i-1}. \quad (5)$$

На основе описанной в соотношении (4) математической модели АФНФ был предложен алгоритм определения стабильности запроса CH на основании изменения младшего и старшего разрядов. Данный алгоритм позволяет оценить вероятность попадания ответа R на запрос CH в регион метастабильности. На рис. 5 показаны результаты параметрического моделирования 16-разрядной АФНФ на ПЛИС Xilinx Artix-7 [38].

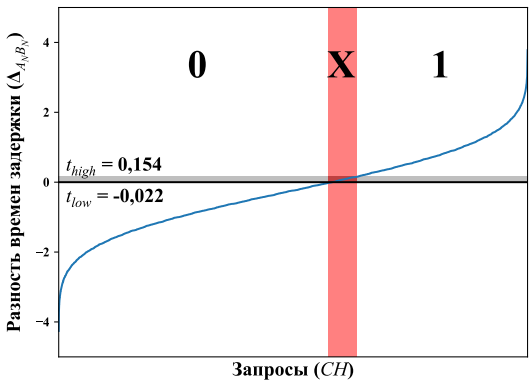

Рис. 5 – Пары запрос-ответ, отсортированные по значению $\Delta A_N B_N$

В результате моделирования показано, что из $2^{16} = 65536$ пар запрос-ответ 67 % являются стабильными, а 33 % – нестабильными. Данный анализ подтверждает недостаток АФНФ с небольшой разрядностью, поскольку значительная часть пар запрос-ответ были классифицированы как нестабильные. На практике при реализации АФНФ разрядностью как минимум 128, даже несмотря на значительный процент отбракованных запросов, мощность множества стабильных пар запрос-ответ достаточна для того, чтобы произвести идентификацию без повторного использования пар запрос-ответ.

Применение алгоритмов, предложенных в работах [39–41] показало, что стабильность АФНФ была улучшена до 1,0. Алгоритм был также верифицирован в условиях изменения температуры окружающей среды от –40 до +90 °C. Результаты тестирования показателя P_{stable} (вероятности стабильности всего множества пар запрос-ответ) приведены на рис. 6.

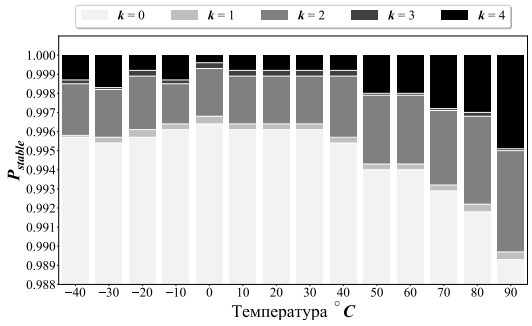

Рис. 6 – Результаты тестов стабильности при изменении температуры

Показатель k обозначает количество бит запроса, измененных для тестирования запроса. Например, если $k = 2$, то изменению подвергаются два младших бита запроса. Следовательно, для обеспечения вероятности 1,0 достаточно изменить в запросе 4 младших бита для определения его стабильности.

Также была экспериментально исследовано 20 идентичных ПЛИС Xilinx Artix-7 и одна

ПЛИС Xilinx ZC706 [42]. В результате было установлено, что расположение региона метастабильности, а, соответственно, и показатель уникальности АФНФ обусловлен как особенностями конкретного кристалла ПЛИС, так и расположением ФНФ на кристалле. Результат эксперимента показан на рис. 7.

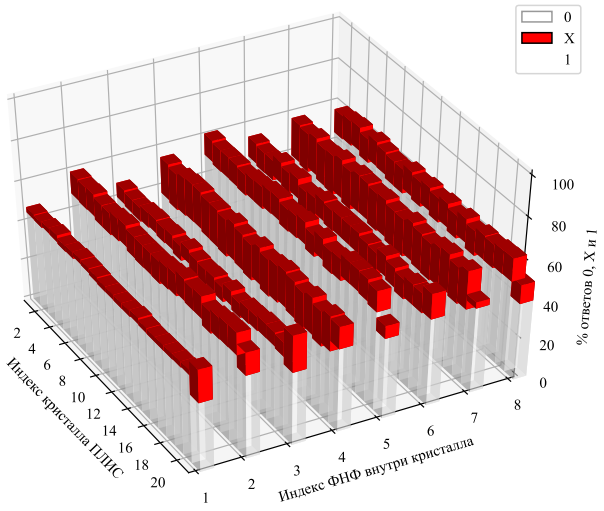

Рис. 7 – Распределение регионов 0, X и 1 для различных кристаллов и компонент АФНФ

Было показано, что генерирование уникального неклонируемого идентификатора цифрового устройства может быть эффективно реализовано с помощью ФНФ типа арбитр на платформе ПЛИС. При этом отсутствие идеальной симметрии путей может быть скомпенсировано за счет модификации схема арбитра, а также применения алгоритмов тестирования запросов с целью определения вероятности их стабильности. Таким образом, стабильность и уникальность генерируемого идентификатора может быть обеспечена в том числе при реализации ФНФ на ПЛИС.

III. Снижение уязвимости к криптографическим атакам с помощью машинного обучения в протоколах аутентификации

Разность времени задержки распространения сигнала по симметричным путям в АФНФ является линейной функцией, как показано ранее. Аналогично соотношению (2) значение задержки Δ может быть представлено как функция от запроса CH. Не ограничивая общности, представим зависимость ответа R ФНФ для идеально симметричного арбитра на основе единственного D-триггера (в этом случае $t_{low} = t_{high} = 0$):

$$R = \begin{cases} 0, & \text{если } \Delta(CH) < 0, \\ 1, & \text{если } \Delta(CH) > 0. \end{cases} \qquad (6)$$

На рис. 8 показана зависимость значения задержки $\Delta(CH)$ в зависимости от отсортированных бинарных значений запросов CH для 16-разрядной АФНФ. Данные для графика получены в результате проведения параметрического моделирования ПЛИС Xilinx ZC706 в среде Vivado. Из рис. 8 видно, что осуществить разделение двух классов (значений логического ноля и единицы) по значению $\Delta(CH)$ возможно с помощью линейной функции (прямой линии).

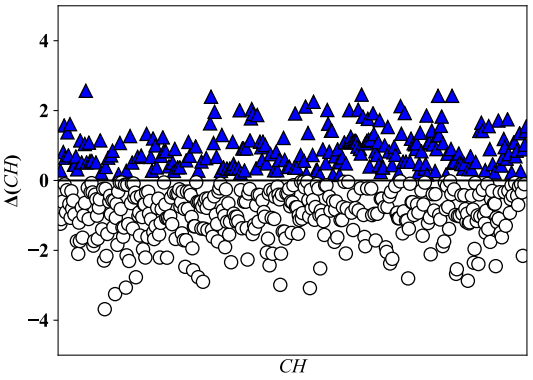

Рис. 8 – Зависимость разности задержек распространения сигнала Δ в зависимости от значения запроса CH

В связи с тем, что разделимость множества пар запрос-ответ очень высокая, многие исследователи показали, что модель, основанная на идеальном арбитре, позволяет правильно предсказать порядка 98–99 % для 64-разрядной АФНФ, имея от 300 до 400 пар запрос-ответ [43], с использованием метода опорных векторов или логистической регрессии.

Авторами предложен [40–41] метод снижения уязвимости к криптографическим атакам с помощью машинного обучения. Данный метод основан на предварительной обработке запросов АФНФ с помощью многоканального сигнатурного анализатора (Multiple Input Signature Register, MISR). MISR позволяет произвести обфускацию запросов АФНФ, тем самым делая зависимость ответов от запросов нелинейной. Более того, основным предназначением блока MISR является вычисление сигнатур для тестирования входных данных, когда ИС спроектирована по принципу тестопригодности (Design For Testability, DFT). Таким образом, если на FPGA реализованы процедуры для самотестирования, то MISR может использован не только для тестирования, но и для обработки запросов АФНФ без дополнительных аппаратурных затрат.

Схемная реализация предлагаемой модификации АФНФ с помощью блока MISR приведена на рис. 9. Как правило, MISR является частью структуры самотестирования BILBO (Built-In Logic Block Observer), которая может быть сконфигурирована в четырех режимах: чтения входных данных, памяти, сдвигового регистра с обратной связью и многоканального сигнатурно-

го анализатора. В предлагаемой реализации используется только два режима: режим памяти ($\alpha_0 = 0$), когда инициализирующее значение загружается в MISR, и режим MISR ($\alpha_0 = 1$) для вычисления запросов, подаваемых на вход АФНФ.

Как показали последние исследования [44], несмотря на отсутствие у злоумышленника прямого доступа к множеству пар запрос-ответ, криптографическая атака может быть проведена с помощью оптимизации методом черного ящика [45]. Наиболее применимым в настоящее время является метод эволюционной стратегии адаптации матриц ковариации (Covariance Matrix Adaptation Evolution Straregy, CMA-ES) [46].

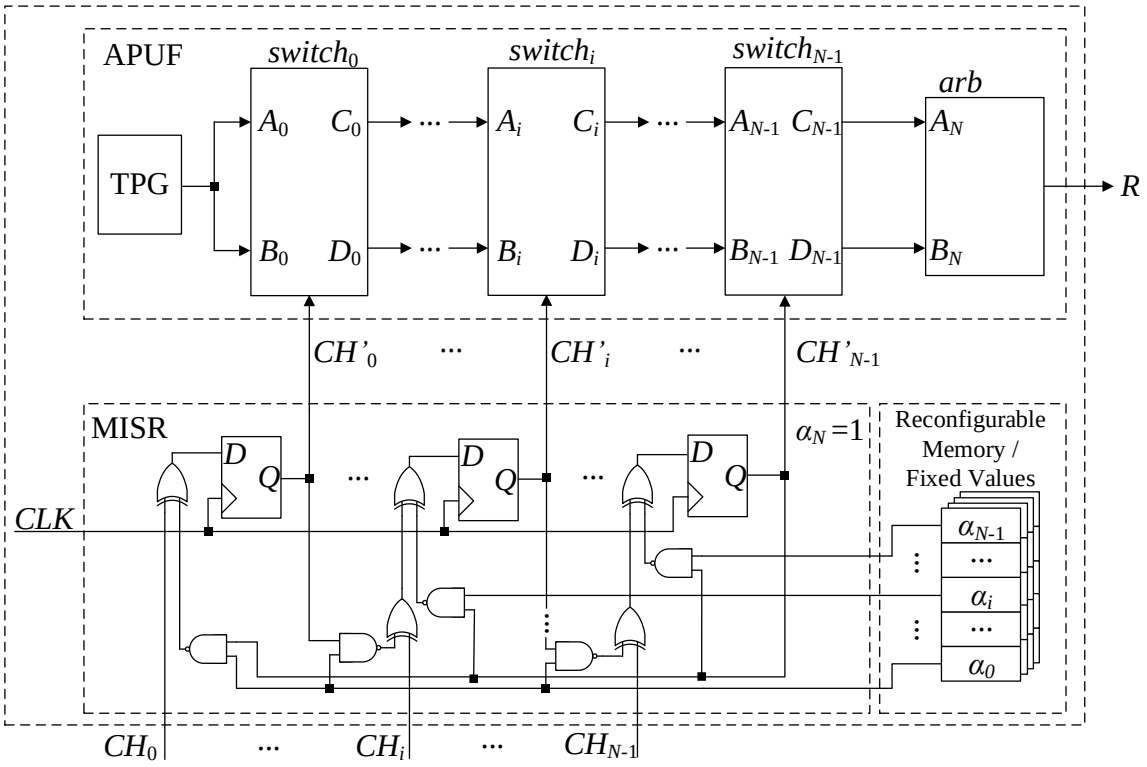

Рис. 9 – Предложенная модификация АФНФ с помощью MISR

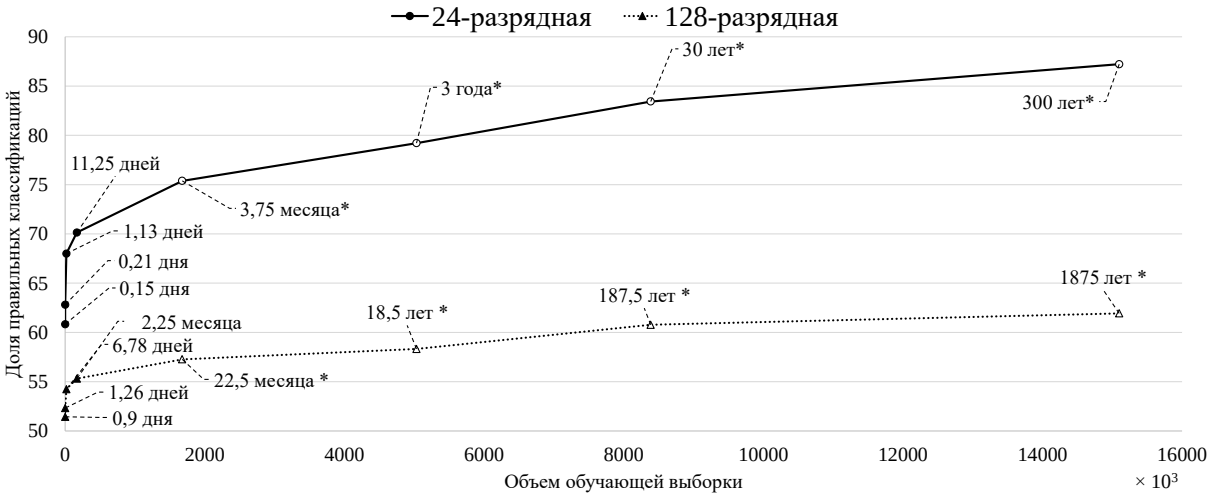

Рис. 10 – Доля успешных классификаций в зависимости от объема обучающей выборки

Данная модификация АФНФ была подвержена криптографическим атакам с помощью методов опорных векторов и эволюционной стратегии. Максимальное значение доли предсказанных значений составило 55 % с применением эволюционной стратегии при этом объем обучаю-

щей выборки составил порядка 10^9 пар запрос-ответ и время обучения – около 3 месяцев. Результат криптографической атаки показан на рис. 10 (точки черного цвета на графике обозначают реальные экспериментальные данные, а точки белого цвета – экстраполированные значения).

Таким образом, применение MISR для обфускации запросов АФНФ позволило снизить практическую уязвимость к криптографическим атакам с помощью машинного обучения, сравнимую с применением алгоритмов хеширования SHA-256 [40–41].

На основе данной модификации АФНФ авторами был предложен протокол аутентификации, общая схема которого приведена на рис. 11.

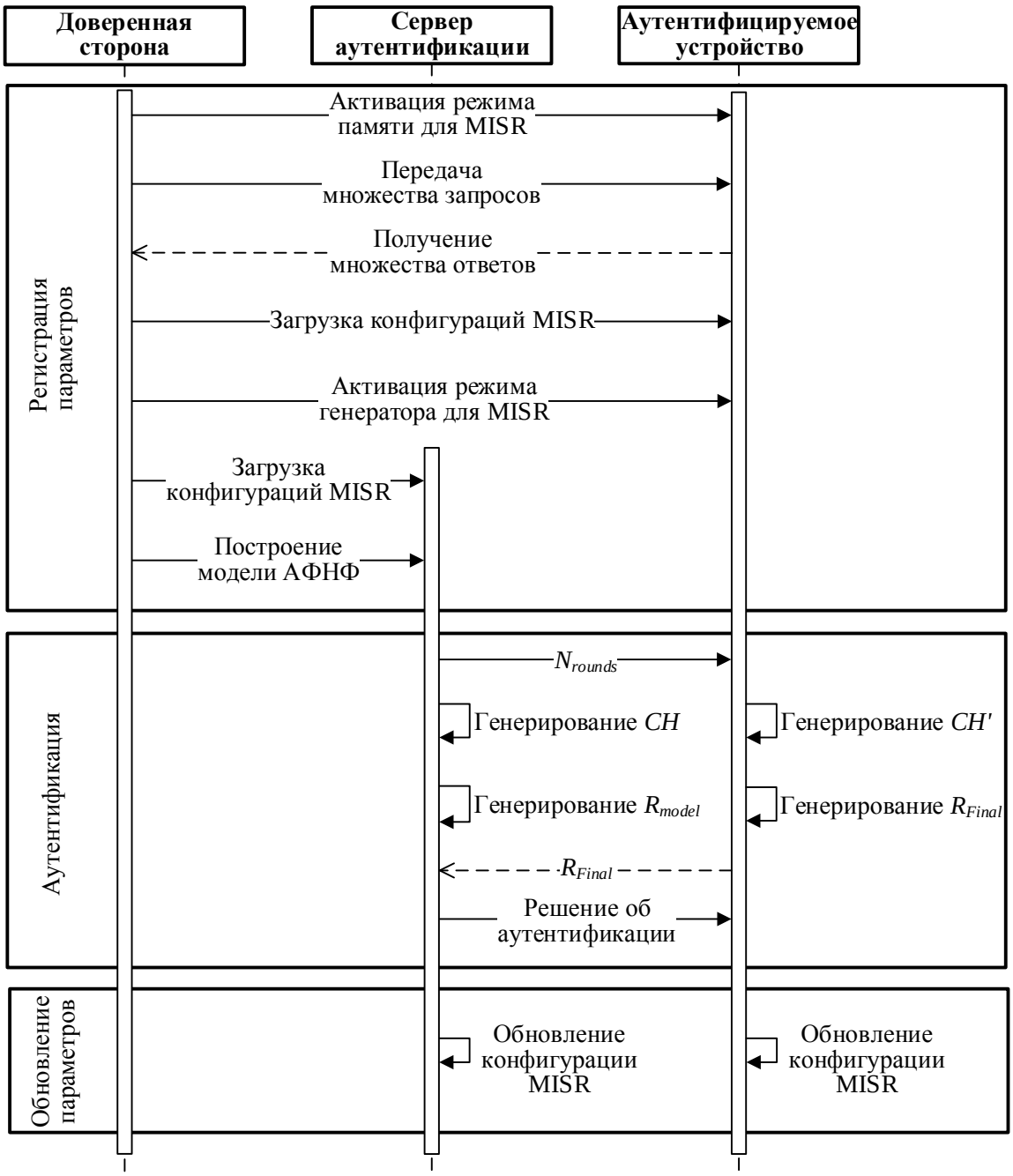

Рис. 11 – Протокол аутентификации на основе модифицированной АФНФ

Предложенный протокол включает в себя три этапа:

1. **Регистрация параметров**. На данном этапе осуществляется инициализация сервера аутентификации и аутентифицируемого устройства путем настройки параметров. На устройстве активируется режим памяти, чтобы запросы не обрабатывались MISR. После чего происходит генерирование множества пар запрос-ответ (например, для 24-разрядной АФНФ требуется несколько миллионов пар, а для 128-разрядной – несколько миллиардов), чтобы построить модель АФНФ с помощью искусственной нейронной сети. Доверенная сторона сохраняет модель АФНФ, а также генерирует параметры MISR (коэффициенты полинома $(_i)$ и начальное значение $(seed)$), которые затем загружаются в реконфигурируемую память аутентифицируемого устройства. Далее происходит активация режима MISR на устройстве, а также передача модели АФНФ и параметров MISR на сервер аутентификации. Таким образом, устройство и сервер готовы к следующему этапу.

2. **Собственно аутентификация**. Процесс аутентификации инициируется устройством. Далее сервер отправляет K случайных значений N_{rounds}, в результате чего на устройстве формируется значение ответа R_{Final}, которое отправляется обратно на сервер. Программная модель АФНФ на сервере используется для вычисления ответа R_{model}, который в силу точности построенной модели должен совпадать со значением R_{Final}. Решение об аутентификации принимается блоком анализатора запросов на сервере на основании равенства или неравенства значений R_{Final} и R_{model}.

3. **Обновление параметров**. С целью обеспечения большей надежности протокола параметры MISR предполагается обновлять не реже раза в месяц. Поскольку количество возможных полиномов в соответствии с функцией Эйлера для 128-разрядного MISR составляет $\approx 1,3 \times 10^{36}$, случайная выборка нескольких из них для периодического обновления позволит избежать криптографической атаки методом исчерпывающего перебора вариантов.

Предложенный протокол позволяет произвести надежную аутентификацию устройства, содержащего АФНФ, без необходимости хранить экспоненциально большое число пар запрос-ответ на сервере, поскольку модель АФНФ потребляет в сотни раз меньше ресурсов памяти для ее хранения.

В отличие от существующих методов построения программных моделей АФНФ, которые позволяют добиться точности от 95 до 98 %, разбиение ответов на четверки и их классификация позволяет добиться стопроцентной точности. Авторами было предложено использовать классификатор, состоящий из трех этапов, которые показаны на рис. 12.

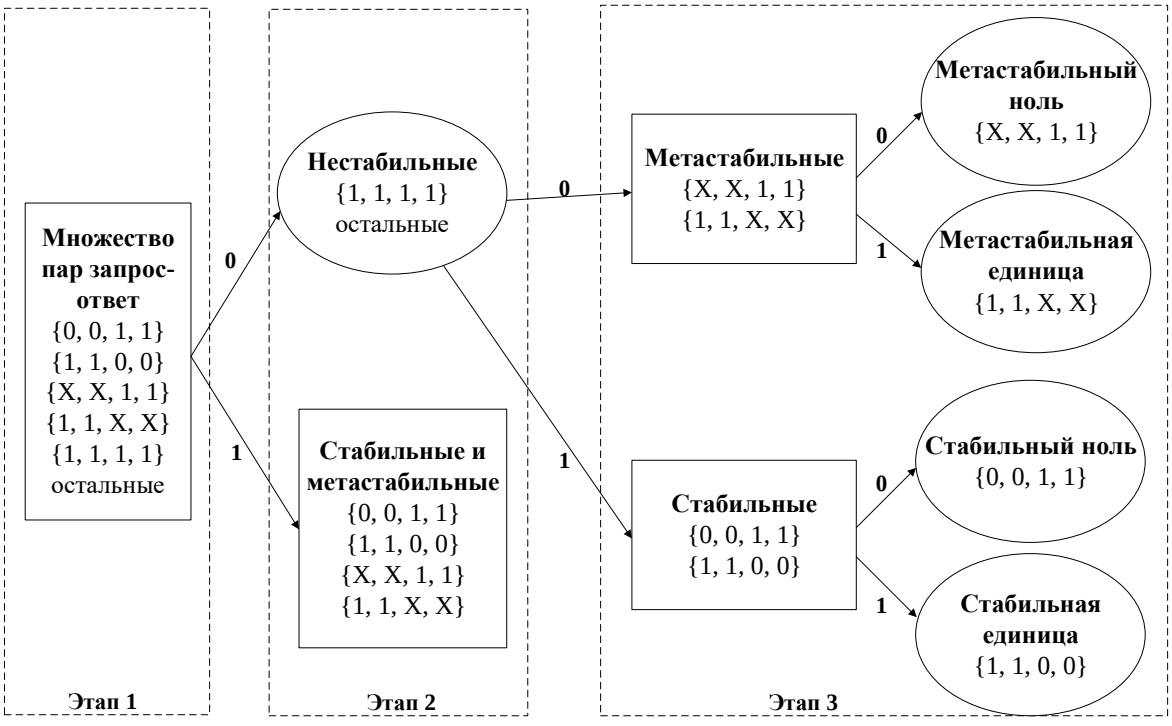

Рис. 12 – Структурная схема алгоритма классификации

Первый этап классификации является наиболее сложным, поскольку алгоритму необходимо отделить стабильные четверки-ответы от нестабильных. Для решения данной задачи была использована глубокая нейронная сеть (Deep Neural Network, DNN), архитектура которой показана на рис. 13.

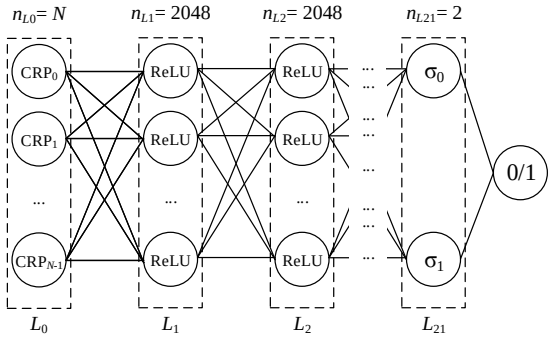

Рис. 13 – Структурная схема алгоритма классификации

Входной слой сети (L_0) состоит из N узлов, каждый из которых является значением знака задержки в запросе ($Sign_i = \{-1, 1\}$, $i = 0..N-1$). Выходы первого слоя далее последовательно подаются на 20 скрытых слоев (L_1, …, L_{20}), каждый из которых состоит из 2048 нейронов с линейно-пороговой функцией активации (Rectified Linear Unit, ReLU). Выходной слой состоит из двух узлов $_0$ и $_1$ с обобщенной логистической функцией активации (Softmax). Таким образом, обученная сеть по значениям знаков задержек выдает ровно два значения: вероятность p_1 того, что это стабильная четверка-ответ ({0, 0, 1, 1}, {1, 1, 0, 0}, {X, X, 1, 1}, {1, 1, X, X}) и вероятность нестабильной четверки p_0 ({1, 1, 1, 1} и другие значения), $p_0 + p_1 = 1$. Таким образом, этап 1 определяет, является ли стабильным запрос, поданный на вход сети.

В данном эксперименте были построены модели двух различных конфигураций АФНФ (24- и 128-разрядная). В каждом случае множество пар запрос-ответ было разбито на три подмножество: обучающая (80 % пар), валидационная (10 %) и экзаменационная выборки. Каждый из запросов был повторно подан на вход АФНФ $E = 100$ раз для определения стабильности каждой из четверок-ответов, наблюдаемых в эксперименте. В соответствии с алгоритмом мажоритарного выбора каждая четверка была помечена в обучающей выборке одним из пяти возможных вариантов: {0, 0, 1, 1}, {1, 1, 0, 0}, {X, X, 1, 1}, {1, 1, X, X}, {1, 1, 1, 1}. Если в результате эксперимента была получена четверка, отличающаяся от перечисленных выше, то данный запрос был отнесен к классу “остальные”. Проблема переобучения предлагаемой сети была решена с помощью регуляризации второго порядка, а также алгоритма сброса весов с вероятностью $p = 0{,}5$. Множество пар запрос-ответ, используемое для обучения сети, содержало 2^{22} и 10^{10} пар для 24- и 128-разрядной АФНФ соответственно. Для каждой реализации АФНФ процент правильных классификаций составил 100 на обучающей, валидационной и тестовой выборках.

Второй этап алгоритма классификации имеет меньшую сложность, поскольку необходимо разделять стабильные ответы-четверки от метастабильных. В связи с этим нейронная сеть вместо 20 слоев содержала только три в отличие от первого этапа. Аналогично нейронной сети, состоящей из 20 слоев, данная модель также продемонстрировала 100 % правильных классификаций как для 24-разрядной, так и для 128-разрядной АФНФ.

Третий этап классификации был реализован с помощью линейного метода классификации (логистической регрессии). Поскольку основная масса четверок-ответов (86,94 %) относится к классам {0, 0, 1, 1}, {1, 1, 0, 0}, то использование простого алгоритма позволит сократить временные издержки как на обучение модели, так и на предсказание метки в процессе аутентификации.

Построение точной модели АФНФ с помощью методов машинного обучения позволило значительно снизить уязвимость предложенного протокола к криптографическим атакам.

IV. Генераторы случайных числовых последовательностей на основе ФНФ

Предложенная авторами [47–48] структура генератора случайных числовых последовательностей (ГСЧП) состоит из трех компонентов: источника случайности, схемы сжатия и регистра случайного числа.

Источник случайности вырабатывает начальную последовательность, которая, как правило, не обладает необходимыми статистическими характеристиками, чтобы использоваться в криптографических приложениях. Схема сжатия применяется в генераторе для улучшения статистических свойств начальной случайной последовательности с целью применения ее в различных приложениях (в том числе криптографических). Регистр случайного числа предназначен для хранения элементов генерируемой случайной последовательности.

Результаты исследования вероятностных характеристик ГСЧП, построенных на основе ФНФ, позволяют сделать вывод о том, что практически любая реализация цифровой ФНФ на базе FPGA применима для генерирования случайных числовых последовательностей высокого качества. Однако всегда стоит принимать во внимание характеристику стабильности, высокое значение которой говорит о том, что ФНФ может быть эффективно использована для идентификации ПЛИС или проекта ПЛИС.

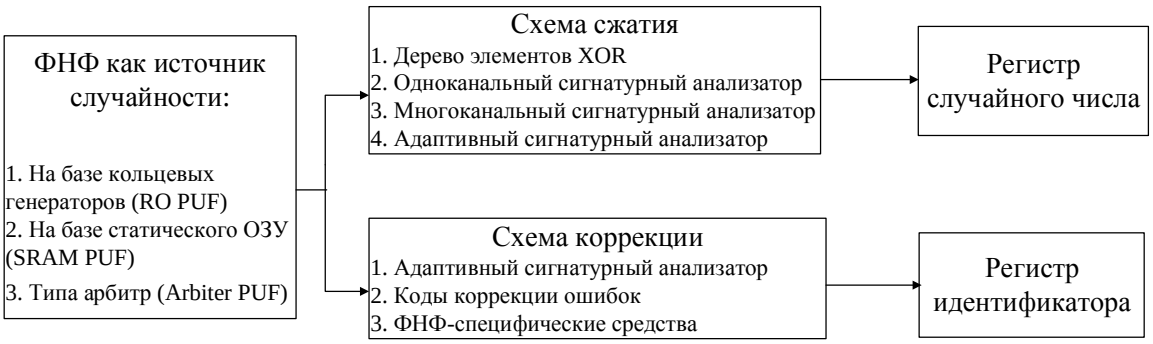

Рис. 14 – Общая структура устройства, работающего в двух режимах: идентификации и генерирования случайных числовых последовательностей

В качестве иллюстрации метода синтеза цифровых устройств на основе ФНФ, поддерживающих режимы идентификации и генерирования случайных числовых последовательностей, на рис. 14 приведена обобщенная структура устройств данного класса. Множество пар запрос-ответ ФНФ может быть условно разделено на два подмножества пар с высокой и низкой стабильностью. Идентификаторы цифрового устройства могут быть построены на основе достаточного числа максимально стабильных пар, которые также обладают высокой характеристикой уникальности. В свою очередь, пары с низкой стабильностью, которые по сути и характеризуют вариации технологического процесса изготовления ИС, следует использовать для генерирования случайных числовых последовательностей.

ФНФ типа арбитр является более эффективной как с точки зрения идентификации, так и ГСЧП в силу простоты реализации и меньших аппаратурных затрат по сравнению с другими классическими ФНФ. По скорости генерирования последовательностей предлагаемые решения ограничены 1 Мбит/с в силу невысокой производительности используемого семейства ПЛИС, а также протокола UART. При реализации данных ГСЧП на ПЛИС Xilinx Zynq-7000 производительность может быть увеличена до 3 Мбит/с при использовании протокола UART и до 100 Мбит/с при использовании Ethernet различных стандартов.

Предложенные авторами ГСЧП успешно проходят тесты из статистических пакетов NIST [49] и Diehard [50], что позволяет сделать вывод о действительной случайности генерируемых ими числовых последовательностей.

V. Экспериментальное исследование физически неклонируемых функций

Авторами предложен общий подход к исследованию характеристик ФНФ (уникальности, случайности, стабильности). Например, для достоверной оценки уникальности требуется как минимум 10 идентичных ИС, содержащих реализацию ФНФ [51]. В связи с этим особенностью реализованной экспериментальной установки является возможность параллельно получать данные с каждой реализации ФНФ.

С другой стороны, реализованная АФНФ обладает расширенным выходным алфавитом за счет обнаружения метастабильных состояний, поэтому алгоритмы подсчета метрик качества были адаптированы для тернарных векторов. Более того, на каждой ПЛИС было реализовано несколько компонент ФНФ, что позволило оценить не только межкристальную, но и внутрикристальную уникальность.

Для исследования характеристик АФНФ была реализована экспериментальная установка, построенная на базе 10 ПЛИС Xilinx Artix-7, входящих в состав плат быстрого прототипирования Digilent Nexys-4, а также сервера Fujitsu PRIMERGY Econel 200 на базе двухъядерного процессора Intel Xeon 5050 с тактовой частотой 3 ГГц и объемом оперативной памяти 8 Гб под управлением операционной системы MS Windows Server 2012. Доступ к серверу осуществлялся удаленно с помощью утилиты Remote Desktop Connection из БГУИР (г. Минск) и НТУ (г. Сингапур). Общая схема эксперимента приведена на рис. 15.

Проектное описание АФНФ было создано на языке VHDL с использованием САПР Xilinx ISE System Edition 14.7. На каждой ПЛИС было реализовано по D компонент мультиарбитральных АФНФ. Разрядность МАФНФ N и, соответственно, количество арбитров задавалось на стадии проектирования. Целью данного эксперимента было повышение стабильности АФНФ за счет использования различных схем арбитра: на основе единственного D-триггера, четырех D-триггеров и асинхронного RS-триггера. Соответственно, тип арбитра также конфигурировался

в исходном коде проектного описания. Создание конфигурационного файла и программирование ПЛИС было осуществлено стандартными средствами САПР (модуль ISE iMPACT).

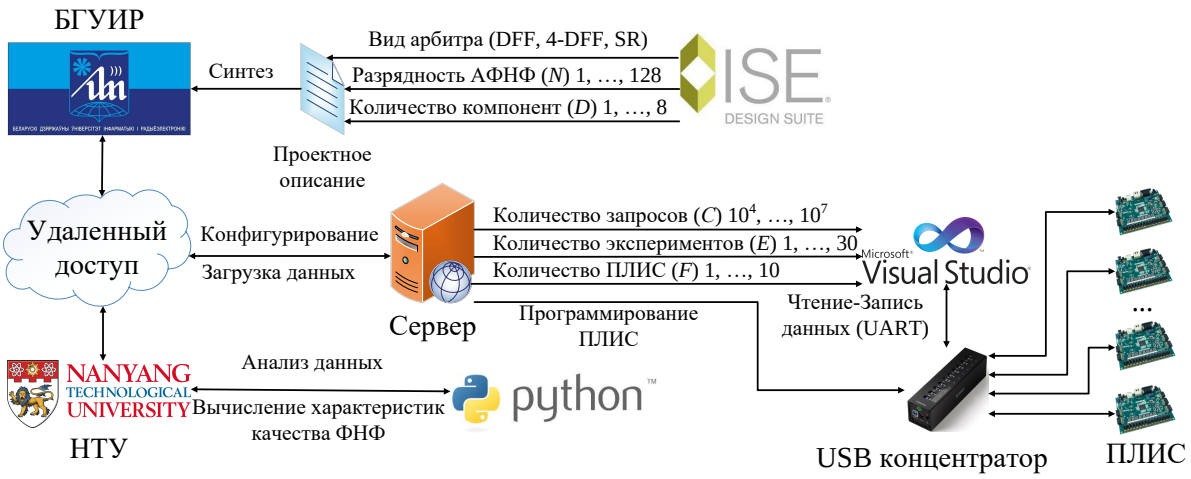

Рис. 15 – Обобщенная схема экспериментальной установки

ПЛИС Xilinx Artix-7 XC7A100T-1CSG324C изготовлена по 45 нм техпроцессу и содержит 15850 секций по шесть четырехвходовых LUT-блоков и восемь синхронных D-триггеров, 4860 Кб встроенной статической памяти (BRAM), четыре блока цифровой обработки сигналов, 300 пользовательских блоков ввода / вывода. Плата быстрого прототипирования Digilent Nexys-4 содержит источник тактового импульса частотой до 450 МГц для синхронизации ПЛИС. Для проведения эксперимента платы были соединены в стойку и подключены к серверу через 10-портовый USB концентратор ST Lab U-500, как показано на рис 16.

Передача данных между сервером и ПЛИС была реализована по протоколу UART. Для реализации получения и передачи данных были использованы стандартные порты UART_TXD_IN, UART_RXD_OUT, UART_CTS, UART_RTS платы Nexys-4. Для генерирования запросов был реализован LFSR (Linear Feedback Shift Register) с разрядностью, совпадающей с количеством звеньев АФНФ (N). Значения запросов являлись слабокоррелированными, поскольку каждый из них вырабатывался только спустя N тактов после предыдущего. Ответы АФНФ, в свою очередь, хранились в регистровом файле.

Соответственно, передача данных от регистрового файла на интерфейс UART реализована в виде отдельного контроллера ФНФ. В свою очередь, программное обеспечение для передачи данных от сервера на АФНФ (ПЛИС) было разработано на языке C# в среде Microsoft Visual Studio. Количество ПЛИС (F), программируемых в данном эксперименте, можно было задать программно, как и число запросов (C). Некоторые из экспериментов были многократно повторены E раз. Данный параметр также задается с помощью программного обеспечения без необходимости перепрограммирования ПЛИС.

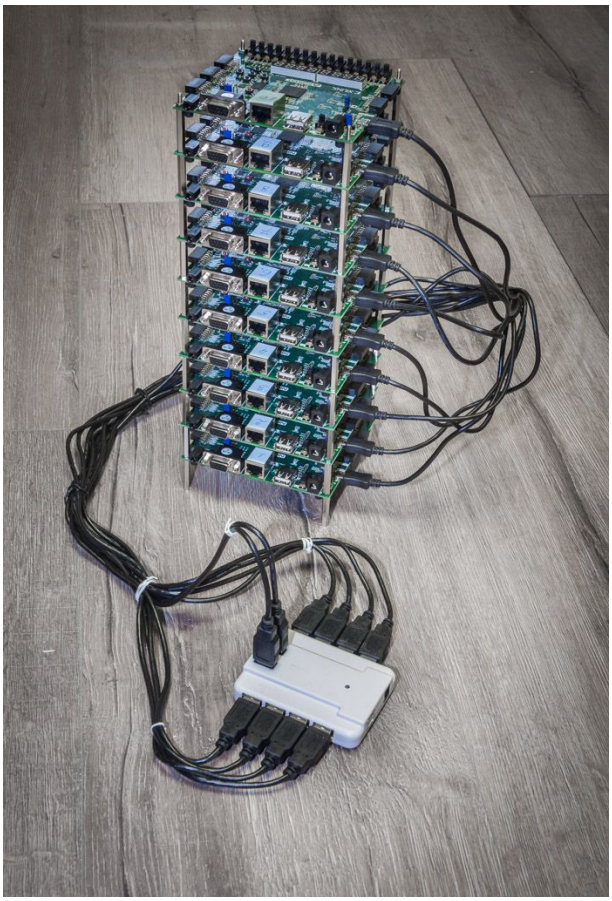

Рис. 16 – Стойка для 10 плат быстрого прототипирования Digilent Nexys-4 и USB концентратор

Данные, полученные в результате эксперимента, сохранялись в текстовые файлы специального формата, а затем анализировались с помощью алгоритмов расчета стабильности и уникальности на языке Python. В силу того, что выходной алфавит АФНФ тернарный, для проверки случайности метастабильные состояния были перекодированы в значения 0 и 1 с помощью генератора псевдослучайных числовых последовательностей, реализованного на языке Python. Пары запрос-ответ АФНФ анализировались с помощью пакетов NIST и Statistica.

Как было показано ранее, характеристика стабильности ФНФ может значительно ухудшаться при изменении условий функционирования ЦУ. В связи с этим метрика стабильности была измерена в условиях изменяющейся температуры окружающей среды от –40° C до 90° C. Исследование показателя стабильности под воздействием различных температур было осуществлено с помощью температурной камеры Thermotron ®8800, доступ к которой был предоставлен Центром исследования спутников Наньянгского технологического университета (Nanyang Technological University Satellite Research Centre) в Сингапуре. На рис. 17 приведена фотография температурной камеры и ПЛИС, находящейся внутри.

Рис. 17 – Температурная камера Thermotron ®8800

Чтение и запись $C = 10000$ запросов с четырьмя возможными комбинациями дополнительного старшего и младшего бит, сгенерированных повторно $E = 100$ раз, эквивалентны обработке $4 \cdot 10^6$ запросов классической АФНФ. Обработка данного количеста запросов занимает порядка 40 минут. Процесс чтения и записи был повторен в диапазоне температур от –40 до 90 °C с шагом 10 °C. Таким образом, проведение эксперимента требует порядка 10 часов.

Программирование температурной камеры представляет собой настройку времени работы при каждой температуре, времени перехода между температурами, а также времени остывания (нагревания) камеры до комнатной температуры. Перечисленные параметры настраиваются с помощью передней панели камеры, управляемой касаниями стилуса, как показано на рис. 18. Конфигурационный файл, содержащий алгоритм изменения температуры, сохраняется в формате *.pgm. После нажатия кнопки Run на панели управления камера автоматически изменяет температуру внутри в соответствии с заданными временными интервалами.

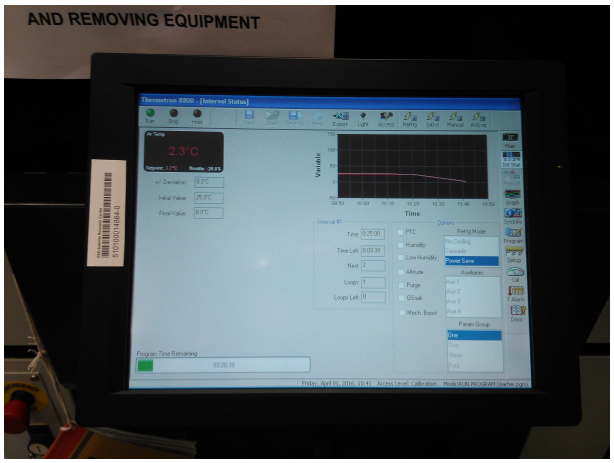

Рис. 18 – Панель настройки температурной камеры

Чтение и запись данных были осуществлены с помощью ноутбука MacBook Pro Mid 2015 на базе процессора Intel Core i5-5257U с объемом оперативной памяти 8 Гб. Кабели, соединяющие ноутбук и плату быстрого прототипирования, были помещены внутрь камеры и отделены от внешней среды с помощью предохранительной мембраны.

Проведение эксперимента в условиях изменяющейся температуры окружающей среды позволило подтвердить гипотезу о том, что устойчиво сильные запросы, выбранные в условиях комнатной температуры, сохраняют высокие характеристики стабильности.

VI. Заключение

В период с 2012 по 2019 годы профессором Иванюком А.А. и доцентом Заливако С.С. было опубликовано 8 статей в рецензируемых на-

учных изданиях, глава монографии в книге издательства Springer, 5 статей на международных научных конференциях (ISIC-2014 – Сингапур; ASPDAC-2016 – Макао, КНР; ISQED-2017 – Санта-Клара, США; ISCAS-2017 – Балтимор, США, PRIP-2019 – Минск, Беларусь) и более 20 тезисов докладов на международных и республиканских научных конференциях.

В настоящее время существует ряд открытых проблем, не решенных международным научным сообществом в области ФНФ: исследование и разработка новых архитектур ФНФ, реализация ФНФ на основе готовых интегральных схем, нахождение компромисса между стабильностью и предсказуемостью ФНФ, разработка точных и универсальных математических моделей ФНФ, возможности для использования ФНФ в коммерческих проектах.

Авторы в настоящее время ведут исследования в области разработки новых архитектур ФНФ на основе флэш-памяти, проектирования ФНФ на платформе ПЛИС с уменьшенными аппаратурными затратами, повышения стабильности и уникальности неклонируемых идентификаторов цифровых устройств, а также снижения уязвимости к криптографическим атакам с помощью машинного обучения.

Интерес международного научного сообщества и ведущих мировых производителей полупроводниковых устройств к тематике ФНФ подтверждает актуальность выбранного авторами научного направления.

VII. Список литературы

1. State of the IoT 2018: Number of IoT devices now at 7B — Market accelerating [Electronic resource] / K. L. Lueth. – IoT Analytics, 2018. – Mode of access: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/ – Date of access: 14.10.2019.
2. Internet of Things forecast [Electronic resource] – Ericsson, 2019. – Mode of access: https://www.ericsson.com/en/mobility-report/internet-of-things-forecast – Date of access: 14.10.2019.
3. Ahmad M., A Review of Current Security Issues in Internet of Things / M. Ahmad [at el] // Springer, 2019. – PP. 11–23.
4. Chang, C. H. and Potkonjak, M. Secure System Design and Trustable Computing / C. H. Chang // Springer, 2016. – 537 P.
5. Иванюк, А. А. Физическая криптография и защита цифровых устройств / А. А. Иванюк, С. С. Заливако // Доклады БГУИР – 2019. – № 2 (120). – С. 50–58.
6. Chang, C. H. A Retrospective and a Look Forward: Fifteen Years of Physical Unclonable Function Advancement / C. H. Chang, Y. Zheng , L. Zhang // IEEE Circuits and Systems Magazine – 2017. – Vol. 17, № 3. – P. 32–62.
7. Ярмолик, В. Н. Физически неклонируемые функции / В. Н. Ярмолик, Ю. Г. Вашинко // Информатика – 2011. – № 2 (30). – С. 92–103.
8. Tuyls, P. Security with Noisy Data / P. Tuyls, B. Skoric, T. Kevenaar. – Springer, 2007. – 344 P.
9. Physical one-way functions / R. Pappu [et al.] // Science. – 2002. – vol. 297, № 5589. – P. 2026–2030.
10. A technique to build a secret key in integrated circuits for identification and authentication applications / J. Lee [et al.] // Int. Symp. VLSI Circuits (VLSI'04). – Honolulu, USA, 2004. — P. 176–179.
11. Anderson, J. H. A PUF design for secure FPGA-based embedded systems / J. H. Anderson // Proc. Asia and South Pacific Design Automat. Conf. 2010 (ASP-DAC'10). – Taipei, Taiwan, 2010. – P. 1–6.
12. Silicon physical random functions / B. Gassend [et al.] // ACM Conf. on Comp. and Comm. Security (CCS'02). – New York, USA, 2002. – P. 148–160.
13. The bistable ring PUF: A new architecture for strong physical unclonable functions / Q. Chen [et al.] // Proc. IEEE Int. Sympos. on Hardw. Orient. Secur. and Trust (HOST'11). – San Diego, USA, 2011. – P. 134–141.
14. Holcomb, D. Initial SRAM state as a fingerprint and source of true random numbers for RFID tags / D. Holcomb, W. Burleson, K. Fu // Conf. RFID Security (RFID'07). – Malaga, Spain, 2007. – P. 1–2.
15. DRAM-based intrinsic physically unclonable functions for system-level security and authentication / F. Tehranipoor [et al.] // IEEE Trans. on Very Large Scale Integr. (VLSI) Syst. – 2016. – № 99. — P. 1–13.
16. The butterfly PUF protecting IP on every FPGA / S. S. Kumar [et al.] // Proc. Int. Workshop Hardware-Oriented Security and Trust (HOST'08). – Anaheim, USA, 2008. – P. 67–70.
17. Uniqueness enhancement of PUF responses based on the locations of random outputting RS-latches / D. Yamamoto [et al.] // Cryptographic Hardware and Embedded Systems (CHES'11). – Nara, Japan, 2011. – P. 390–406.
18. Highly reliable memory-based physical unclonable function using spintransfer torque MRAM / L. Zhang [et al.] // IEEE Int. Symp. on Circ. and Syst. (ISCAS'14). – Melbourne, Australia, 2014. –P. 2169–2172.
19. Read-proof hardware from protective coatings / P. Tuyls [et al.] // Cryptographic Hardware and Embedded Systems (CHES'06). – Yokohama, Japan, 2006. – P. 369–383.
20. Zheng, J. X. A digital PUF-based IP protection architecture for network embedded systems / J. X. Zheng M. Potkonjak // Proc. ACM/IEEE Symp. on Archit. for Netw. and Comm. Syst. (ANCS'14). – Marina Del Rey, USA, 2014. – P. 255–256.
21. Efficient Implementation of True Random Number Generator Based on SRAM PUFs / V. Leest [et al.] // Cryptography and Security: From Theory to Applications / ed. by D. Naccache. – Springer, 2012. – P. 300–318.
22. Zalivaka, S.S. Arbiter PUF based FPGA chip identification and authentication methods with enhanced reliability and modeling attack resistance: PhD Thesis. – Singapore: NTU, 2018. – 162 P.
23. Bolotnyy, L. Physically unclonable function-based security and privacy in RFID systems / L. Bolotnyy, G. Robins // Proc. IEEE Int. Conf. on Perv. Comp. and Comm. (PerCom'07). – White Plains, USA, 2007. – P. 211–218.
24. Zhang, X. On-chip structures and techniques to improve the security, trustworthiness and reliability of integrated circuits / X. Zhang [Electronic resource]. — Mode of access: http://digitalcommons.uconn.edu/cgi/viewcontent.cgi?article=6219&context =dissertations. – PhD Thesis, 2013. – Date of access: 14.10.2019.

25. Maes, R. PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator / R. Maes, A. V. Herrewege , I. Verbauwhede // Proc. Crypt. Hardw. and Emb. Syst. (CHES'12). – White Leuven, Belgium, 2012. – P. 302–319.
26. Verayo. Security and authentication solutions based on silicon physical unclonable functions (PUF) technology [Electronic resource]. – Mode of access: http://www.verayo.com/. – 2013, Verayo, Inc. – Date of access: 14.10.2019.
27. Samsung Introduces Exynos i T100 for Secure and Reliable IoT Devices with Short-Range Connectivity [Electronic resource] – Samsung, 2019. – Mode of access: https://news.samsung.com/global/samsung-introduces-exynos-i-t100-for-secure-and-reliable-iot-devices-with-short-range-connectivity – Date of access: 14.10.2019.
28. Altera Partners with Intrinsic-ID to Develop World's Most Secure High-End FPGA [Electronic resource] – Intel, 2015. – Mode of access: https://newsroom.intel.com/news-releases/altera-partners-intrinsic-id-develop-worlds-secure-high-end-fpga/ – Date of access: 14.10.2019.
29. Xilinx to add PUF security to Zynq devices [Electronic resource] – EE News, 2016. – Mode of access: https://www.eenewseurope.com/news/xilinx-add-puf-security-zynq-devices-0 – Date of access: 14.10.2019.
30. Applying circuit delay-based physically unclonable functions (PUFs) for masking operation of memory-based PUFs to resist invasive and clone attacks [Electronic resource] – Google Patents, 2013. – Mode of access: https://patents.google.com/patent/US9787480 – Date of access: 14.10.2019.
31. SRAM PUF Solutions [Electronic resource] – Intrinsic ID, 2019. – Mode of access: https://www.intrinsic-id.com/solutions/ – Date of access: 14.10.2019.
32. Accenture and Thales Demonstrate How Blockchain Technology Can Secure and Simplify Aerospace & Defense Supply Chains [Electronic resource] – Accenture, 2018. – Mode of access: https://newsroom.accenture.com/news/accenture-and-thales-demonstrate-how-blockchain-technology-can-secure-and-simplify-aerospace-and-defense-supply-chains.htm – Date of access: 14.10.2019.
33. Zalivaka, S. S. Multi-valued arbiters for quality enhancement of PUF responses on FPGA implementation / S. S. Zalivaka, A. V. Puchkov, V. P. Klybik, A. A. Ivaniuk, C. H. Chang // Proc. of IEEE Asia and South Pacific Design Automation Conf. (ASP-DAC'2016), – Macau, China – P. 533–538.
34. Kacprzak, T. Analysis of oscillatory metastable operation of an RS flip-flop / T. Kacprzak // IEEE J. of Solid-State Circ. – 1988. – Vol. 23, № 1. – P. 260–266.
35. Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications / S. Al-Janabia [et al.] // Egyptian Informatics J. – 2017. – Vol. 18, № 2. — P. 113–122.
36. Highly-reliable feed through/filter capacitor and method for making same [Electronic resource]. – Mode of access: https://www.google.com/patents/US4424551. – 1991. – Date of access: 15.10.2019.
37. Stacey, W. M. Nuclear Reactor Physics / W. M. Stacey. – Wiley, 2007. – 707 p.
38. Nexys 4 DDR artix-7 FPGA: Trainer board recommended for ece curriculum [Electronic resource]. – Mode of access: http://store.digilentinc.com/nexys-4-ddr-artix-7-fpga-trainer-board-recommendedfor-ece-curriculum/. – Digilent, Inc, 2017. – Date of access: 15.10.2019.
39. Заливако С. С., Иванюк А. А., Клыбик В. П. Метод увеличение стабильности физически неклонируемой функции типа "арбитр" // Информатика. 2017. № 1 (53). – С. 31–43.
40. Zalivaka S. S., Ivaniuk A. A., Chang C. H. FPGA Implementation of Modeling Attack Resistant Arbiter PUF with Enhanced Reliability // Invited Paper at Special Session on IoT Security: Protocol, Implementation and Attacks, in Proc. 18st IEEE International Symposium on Quality Electronic Design (ISQED'17). Santa Clara, CA, USA, 13–15 March 2017. – P. 313–318.
41. Zalivaka S. S., Ivaniuk A. A., Chang C.-H. Reliable and Modeling Attack Resistant Authentication of Arbiter PUF in FPGA Implementation With Trinary Quadruple Response // IEEE Transactions on Information Forensics and Security. 2018. № 4 (14). – P. 1109–1123.
42. 7 series FPGAs configurable logic block [Electronic resource]. – Mode of access: https://www.xilinx.com/support/documentation/user_-guides/ug474_- 7Series_CLB.pdf. – Xilinx, Inc, 2016. – Date of access: 15.10.2019.
43. Theory and application of delay constraints in arbiter PUF / U. Chatterjee [et al.] // ACM Trans. on Emb. Comp. Syst. – 2016. – Vol. 15, № 1. – P. 1001–1020.
44. Efficient online and offline testing of embedded DRAMs / S. Hellebrand [et al.] // IEEE Trans. on Comp. – 2002. – Vol. 51, № 7. – P. 801–809.
45. Vicente, L. N. Implicitly and densely discrete black-box optimization problems / L. N. Vicente // Optimization Letters. – 2009. – Vol. 3, № 3. – P. 475–482.
46. Omidvar, M. N. A comparative study of CMA-ES on large scale global optimisation / M. N. Omidvar, X. Li // Australasian Joint Conf. on Art. Intell. – Adelaide, Austalia, 2010. – P. 303–312.
47. Заливако С. С., Иванюк А. А. Использование физически неклонируемых функций для генерирования действительно случайных числовых последовательностей // Автоматика и вычислительная техника. – 2013. № 3. – С. 61–72.
48. Заливако С. С., Иванюк А. А. Схемная реализация комбинированной физически неклонируемой функции для генерирования действительно случайных числовых последовательностей // Докл. БГУИР. – 2013. № 7 (77). – С. 37–43.
49. A statistical test suite for random and pseudorandom number generators for cryptographic applications [Electronic resource]. – Mode of access: http://csrc.nist.gov/groups/ST/toolkit/rng/documents/SP800-22rev1a.pdf. – NIST, 2010. – Date of access: 15.10.2019.
50. Diehard: A battery of tests of randomness [Electronic resource]. – Mode of access: http://stat.fsu.edu/_-geo. – Florida State University, 1995. – Date of access: 15.10.2019.
51. Vijayakumar, A. On testing physically unclonable functions for uniqueness / A. Vijayakumar, V. C. Patil, S. Kundu // Proc. Int. Symp. on Qual. Electr. Design (ISQED'2016). – Santa Clara, USA, 2016. – P. 368–373.