АППАРАТНО-ПРОГРАММНОЕ СРЕДСТВО ТЕСТИРОВАНИЯ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ

Ахмед А.Н.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Иванюк А.А. – д-р техн. наук, профессор

В тексте доклада описывается программное средство тестирования физически неклонирумых функций. Описываются экспериментальная установка, процесс сбора данных, структура хранимых данных и способ анализа получаемых данных.

Физически неклонируемая функция — это аппаратное устройство, экземпляры которого имеют ряд уникальных параметров и характеристик, которые можно применить для генерации пар “запрос-ответ”, с помощью которых можно идентифицировать устройство. Использование ФНФ в криптографии основано на том, что практически невозможно физически скопировать ФНФ и сложно предсказать ответ на определенный запрос.

В силу изменения температуры окружающей среды, а также неизбежного износа и деградации кристалла интегральной схемы, генерируемые ФНФ идентификаторы являются нестабильными. С другой стороны, ФНФ являются хорошим источником случайности для построения на их основе генераторов случайных числовых последовательностей, однако их вероятностные характеристики не всегда соответствуют криптографическим стандартам. В свою очередь, увеличение стабильности ФНФ приводит к уменьшению ее случайности, что способствует уязвимости к криптографическим атакам. Для определения особенностей конкретной ФНФ необходимо вычислить следующие характеристики [3]: случайность (randomness), устойчивость (steadiness), правильность (correctness), диффузия (diffuseness), уникальность (uniqueness). При тестировании любого нового устройства ФНФ необходимо разработать протокол общения с ФНФ, определиться, как и в каком формате будут храниться собранные во время теста данные, выбрать инструмент для анализа полученных данных и визуализации результатов.

Разрабатываемое программное средство тестирования ФНФ должно решать задачу автоматизации вышеописанных действий.

Процесс тестирования можно разделить на два обособленных подпроцесса. Первый подпроцесс — сбор и хранение данных, второй — анализ данных.

На рисунке 1 представлена структура экспериментальной установки, потоки данных между хостом и установкой и внутри установки, которую будем использовать для тестирования. ФНФ примем за черный ящик, программному средству будет необходимо предоставить информацию о разрядности запросов *CH* (challenge) и ответов *R* (response). Тестируемые ФНФ чаще всего реализуют на программируемых логических интегральных схемы (FPGA). Это позволяет нам на той же ИС, на которой реализована ФНФ, реализовать и интерфейс для доступа к ФНФ.

При разработке данного программного средства будут проанализированы ФНФ, реализованные на FPGA Zybo Z7 и Nexys 4 от компании Digilent.

Для того чтобы программное средство могло отправлять запросы на генерацию данных и получать поток сгенерированных ответов, необходимо реализовать на встроенной процессорной системе генератор запросов и модуль, который будет получать и отправлять данные по UART. Для разработки на Zybo Z7 воспользуемся IP-ядром ZYNQ7 Processing System, предназначенным для объединения процессора ARM Cortex-A9 с программируемой логикой FPGA. В свою очередь, интерфейс на Nexys 4 разработаем с использованием MicroBlaze — семейства 32-разрядных микропроцессорных ядер, реализуемых на основе FPGA фирмы Xilinx. Использование данных IP-ядер даст возможность, вместо разработки и реализации конечного автомата, использовать программу, написанную на языке C, которая будет исполняться встроенным процессором.

Общение с FPGA будет происходить с использованием последовательного порта (COM-порта). Выбор данного интерфейса обусловлен простотой этого интерфейса с физической точки зрения, тем, что на FPGA можно воспользоваться уже реализованным интерфейсом UART и на любом компьютере USB-порт может использоваться для эмуляции COM-порта. Количество подключаемых устройств можно будет увеличить, подключив USB-хаб.

Для генерации запросов будет использован генератор М-последовательностей, так как с его помощью можно сгенерировать псевдослучайную двоичную последовательность. Данный вид генераторов является периодическим с периодом $N = 2^n - 1$, где n — длина регистра, с помощью которого формируются значения. Любые комбинации символов длины n на длине одного периода М-последовательности встречаются не более одного раза. В общем для тестирования подойдет генератор любой последовательности, состоящей из уникальных запросов CH.

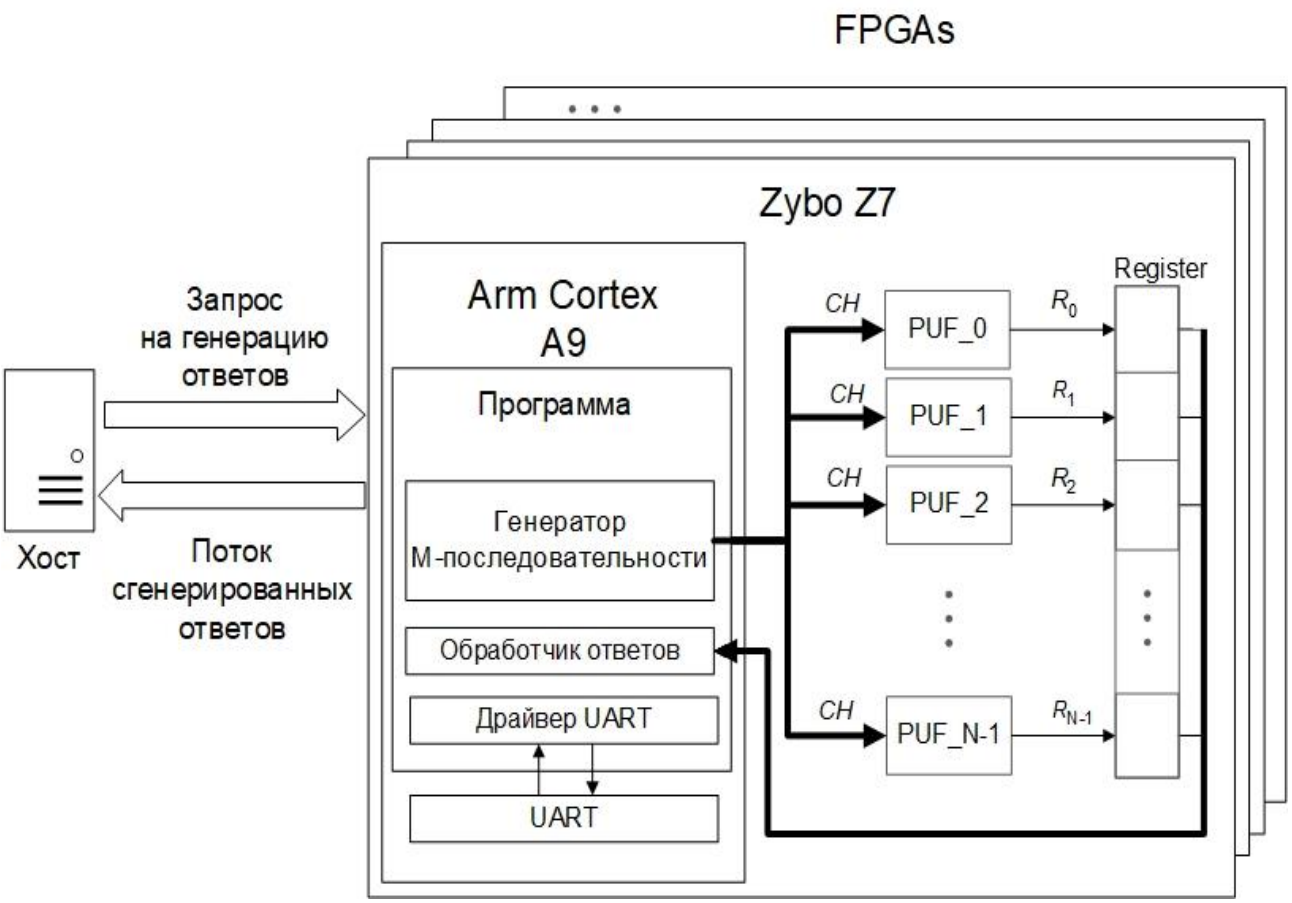

Рисунок 1 – Потоки данных в экспериментальной установке

Полученные ответы будут храниться в файлах. Метаданные об ответах, позволяющие их идентифицировать, будут храниться в базе данных SQLite, так как она не требует от пользователя установки или запуска в контейнере. В ней будут храниться таблицы с информацией об тестируемых ФНФ, названиями её экземпляров, реализованных на FPGA, названиями тестов и дополнительной информацией о них, путями к файлам с ответами.

В файлах будут последовательно записываться ответы, без соответствующего им запроса, это позволит экономить пространство на диске и время на чтение из файла данных и записи данных в файл. Кроме того, вычисление характеристик происходит с использованием порядковых номеров запросов. Ответы экземпляра ФНФ на последовательность запросов хранятся в одной последовательности. Порядок ответов соответствует порядку запросов, из которых они были получены. Для чтения данных из файла необходимо только знать длину ответа в байтах.

В качестве языка разработки программного средства выбран Python, так как он предоставляет широкий набор библиотек для работы со статистикой и числами.

Как отмечается в выводе статьи [1]: для вычисления тех же характеристик надежности и непредсказуемости ФНФ существует множество различных формул. Некоторые из этих формул могут быть алгебраически преобразованы друг в друга, что означает, что они несут одинаковую информацию о характеристике ФНФ, несмотря на разные числовые результаты.

В связи с этим, для гибкости анализа полученных данных, пользователь вместо того, чтобы использовать жестко закодированную логику обработки данных, будет описывать логику обработки математическими выражениями. Таким образом в программное средство будет частично реализовывать функционал таких систем, как Mathcad или Mathematica.

Аппаратно-программное средство тестирования ФНФ значительно повысит эффективность тестирования за счет автоматизации рутинных процессов, что позволит сократить время и улучшить качество работы. А описание логики обработки данных с использованием математических выражений позволит более эффективно модифицировать эту логику.

Список использованных источников:

1.Florian, K. A. Wilde. Metrics for Physical Unclonable Functions / Florian K. A. Wilde – Munich, Technical University of Munich, 2021 – P. 16

2.Maes, R. Physically Unclonable Functions: Constructions, Properties and Applications. PhD thesis / Maes, R – Belgien. Katholieke Universiteit Leuven, 2012.

3. Y. Hori, T. Yoshida, T. Katashita and A. Satoh, "Quantitative and Statistical Performance Evaluation of Arbiter Physical Unclonable Functions on FPGAs," 2010 / Y. Hori, T. Yoshida, T. Katashita and A. Satoh – International Conference on Reconfigurable Computing and FPGAs, Cancun, Mexico, 2010