доступ к каким-либо данным или отдельной функциональности и может быть задействован одновременно в различных приложениях. Таким образом, повышается стабильность системы в целом, улучшается кроссплатформенность, снижается стоимость и повышается скорость разработки.

Поддержка сервис-ориентированной архитектуры очень важна для текущих и будущих проектов на платформе Ruby on Rails, т.к. в современных веб-приложениях конкретные модули и сервисы могут располагаться удаленно, т.е. вне сервера с самим приложением. В настоящий момент Ruby on Rails имеет некоторые ограничения в области сервис-ориентированной архитектуры. Среди них – отсутствующая функциональность для вызова и предоставления сервисов как в самой среде Rails по умолчанию, так и в сторонних решениях в виде одного модуля. В результате появляется необходимость пользоваться несколькими сторонними решениями, которые, ко всему прочему, имеют плохую интеграцию между собой.

Решением данной проблемы может стать разработка единой системы, обеспечивающей расширяемый механизм вызова и предоставления удаленных сервисов по протоколу HTTP и HTTPS для Ruby on Rails. Исходя из требований к разрабатываемой системе, она должна состоять из двух модулей:

1. ServiceInvoker – выполняет запросы к удаленным сервисам и обработку ответов, обеспечивает механизм, который может быть использован для реализации любого протокола, выполняющего запросы к удаленным сервисам через HTTP.
2. ServiceProvider – выполняет прием запросов от удаленных сервисов и их обработку, улучшает существующие возможности среды Rails для предоставления сервисов по протоколу HTTP.

Данная система встраивается в платформу Ruby on Rails в виде подключаемой библиотеки и используется для связи между собой распределенных по сети ресурсов. Достоинством использования данной системы является повышение модульности и стабильности веб-приложения в целом, улучшение его кроссплатформенности, дальнейшее снижение стоимости и повышение скорости разработки.

ПРОГРАММНО-АППАРАТНОЕ СРЕДСТВО ИССЛЕДОВАНИЯ СВОЙСТВ ФИЗИЧЕСКИ НЕКЛОНИРУЕМОЙ ФУНКЦИИ

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

Пучков А. В.

Иванюк А. А. – доктор технических наук, профессор

Одной из ключевых проблем, возникающих при проектировании цифровых устройств, в том числе реализуемых на основе программируемых логических интегральных схем, на современном этапе развития технологий и рынка, является их идентификация. Перспективным направлением разрешения обозначенной проблемы является применение так называемых физически неклонируемых функций. Рассмотренное в работе программно-аппаратное средство даёт возможность провести экспериментальные исследования, позволяющее сделать выводы об эффективности решения задачи идентификации с помощью физически неклонируемых функций для цифровых устройств на базе программируемой логики.

Методы физической криптографии, в основе которой лежит структурная сложность электронных систем, всё чаще находят применение в области защиты цифровых систем от нелегального использования [1]. Согласно одному из современных формальных определений, которое было предложено П. Туилсом (P. Tuyls), физически неклонируемая функция (англ. Physically Unclonable Function, PUF) есть характеристика физической (цифровой) системы, которая не подлежит клонированию (копированию, воспроизведению) на других системах [2]. В процессе создания цифровых систем принципиально невозможно управлять величинами многих физических параметров, вследствие чего последние из-за физической вариации технологического процесса принимают случайные, но уникальные для каждой цифровой системы значения. Идея извлечения подобных уникальных параметров из цифровых систем и лежит в основе аппаратных PUF [1].

Задача PUF как цифрового устройства состоит в получении на выходных портах множества ответов R, соответствующих множеству запросов C таким образом, что пары (C_i, R_i) будут уникальными, непредсказуемыми и неклонируемыми на других аналогичных интегральных схемах [3]. Такие свойства позволяют использовать PUF для эффективного решения целого ряда задач, в первую очередь для уникальной идентификации цифровых систем.

Основой построения и функционирования многих PUF является измерение задержки сигналов в реконфигурируемых путях цифровых устройств. Классическим примером такого подхода является PUF типа арбитр, где на одном кристалле интегральной схемы осуществляется построение двух топологически и функционально идентичных путей, имеющих близкие, но принципиально различные из-за физической вариации технологического процесса, величины времени распространения сигналов по ним. Симметричные пути проектируются как пары двухвходовых мультиплексоров, управляющие входы которых образуют шину входных запросов PUF. Измерение разницы во времени распространения сигналов между двумя путями может быть достигнуто одновременной подачей фронта импульса и определением на выходах, какой из них оказался длиннее [1]. Последнее может в простейшем случае быть достигнуто при помощи синхронного D-триггера, сбрасываемого до выполнения измерений. В этом случае один из путей поступа-

ет на вход данных D-триггера, а другой – на его вход синхронизации. Очевидно, что в данном случае ответ, снимаемый с выхода D-триггера будет показывать, какой из путей имеет большую задержку распространения сигнала.

В рассматриваемом программно-аппаратном средстве была использована принципиально модифицированная схема, основанная на классической реализации PUF типа арбитр – мультиарбитральная PUF (рис. 1), где арбитры подключены к каждому звену симметричных путей. Выбор одного из сигналов ответов осуществляется с помощью мультиплексора [3].

Данная реализация является очень гибкой и позволяет провести экспериментальное исследование свойств рассматриваемой PUF. К характеристикам, доступным для наблюдения, относятся, например, зависимости ответов от одного арбитра при различных значениях входных запросов, ответов от разных арбитров при одном и том же входном запросе, а также их комбинации. При этом принципиально важно, что в некоторых случаях ответы PUF могут быть нестационарными, что вызвано переходом арбитров в метастабильное состояние. Важными свойствами также является зависимость характеристик от параметров окружающей среды, саморазогрева кристалла и напряжения питания, что также доступно для экспериментального исследования и визуализации.

Для реализации мультиарбитральной PUF в рамках рассматриваемого программно-аппаратного средства были выбраны программируемые логические интегральные схемы типа FPGA Xilinx Spartan-3E XC3S500E-5FG320, представленные платами быстрого прототипирования Digilent Nexys 2. Для данной серии FPGA были разработаны проектные описания на языке VHDL мультиарбитральной PUF, контроллера, генераторов входных запросов, а также интерфейсного модуля, осуществляющего взаимодействие с рабочей станцией. Подчинёнными устройствами контроллера являются генераторы входных запросов, множественные компоненты мультиарбитральной PUF, а также интерфейсный модуль. В качестве интерфейса передачи данных между FPGA и рабочей станцией был выбран USB, на основе которого работает протокол DSTM (Digilent Synchronous Parallel Interface). В данной конфигурации со стороны рабочей станции в пакетном режиме поступают команды внутреннего контроллера, которые буферизируются посредством двухпортовой памяти типа FIFO. В свою очередь контроллер обрабатывает команды, вырабатывая управляющие воздействия на генераторы входных воздействий, множественные компоненты мультиарбитральной PUF, а также регистрируя ответы.

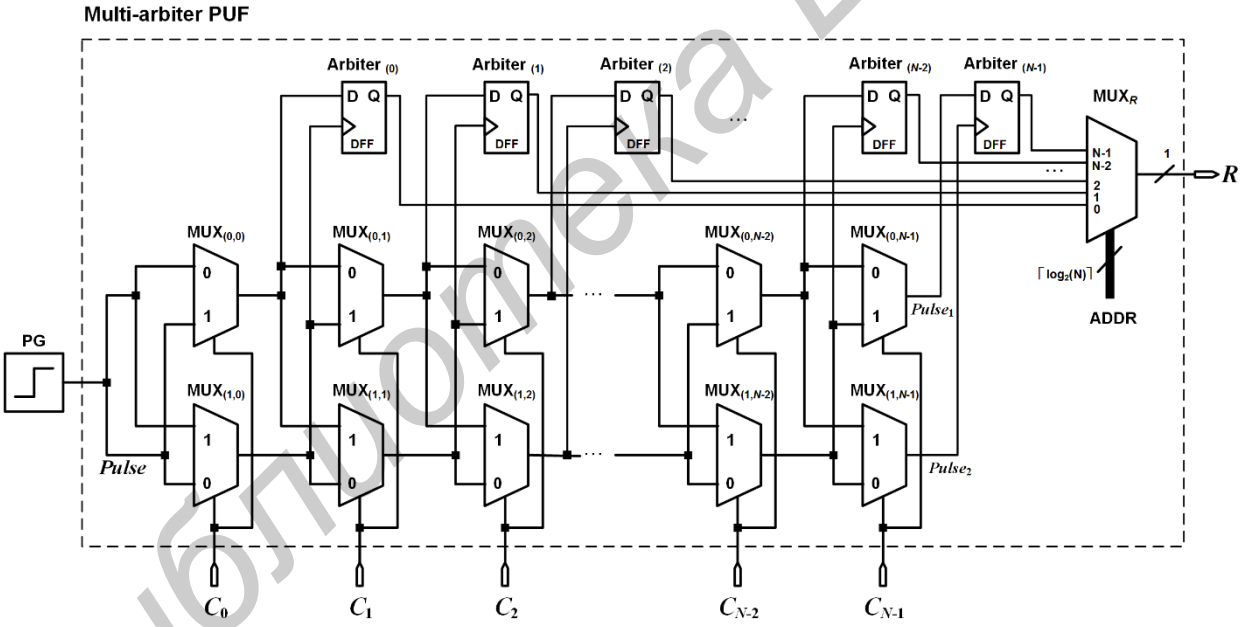

Рис. 1 – Функциональная схема мультиарбитральной PUF

Ответы PUF буферизируются, и по команде с рабочей станции могут с помощью пакетной передачи быть получены клиентской стороной, представленной программным средством.

Клиентское программное средство, выполняющееся на рабочей станции, разработано на языке C++ с помощью интегрированной среды разработки Microsoft Visual Studio 2012 Ultimate. Оно имеет расширяемую архитектуру, что позволяет легко адаптировать его к изменяющимся требованиями постановки эксперимента, а также различным интерфейсам и протоколам взаимодействия с аппаратными реализациями PUF. Для достижения последнего код, обеспечивающий передачу данных между клиентским программным средством и ПЛИС, размещён в отдельных модулях, представляющих слой аппаратных абстракций. Для реализации интерфейсного взаимодействия с помощью протокола DSTM был использован набор для разработки Digilent Adept SDK.

В результате своей работы клиентское программное средство создаёт пригодные для визуализации файлы с данными ответов PUF, полученных от различных входных запросов.

В ходе первичных экспериментов на рассматриваемом программно-аппаратном средстве была экспериментально показана целесообразность и эффективность использования мультиарбитральной PUF

для решения проблемы уникальной идентификации как внутри отдельно взятой интегральной схемы, так и между интегральными схемами.

Список использованных источников:
1. Иванюк, А. А. Проектирование встраиваемых цифровых устройств и систем: монография / А. А. Иванюк. – Минск: Бестпринт, 2012. – 337 с.
2. Tuyls, P. Security with Noisy Data / P. Tuyls, B. Skoric, T. Kevenaar. – London: Springer, 2007. – 344 p.
3. Клыбик В. П., Иванюк А. А. Применение физически неклонируемой функции типа арбитр для решения задачи идентификации цифровых устройств. / В. П. Клыбик, А. А. Иванюк – Информатика, 2015. – в печати.

ЕДИНЫЙ ЦЕНТР ОБРАБОТКИ УДАЛЁННЫХ КОМАНД

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

Скакун А. С.

Бахтизин В. В. – кандидат. техн. наук, доцент, профессор

В связи с распространением «облачных» технологий и сервисов возникает частая необходимость разработки методов взаимодействия между клиентом и сервером, что вынуждает разработчиков тратить дополнительное время при разработке ПО. Наличие какого-либо универсального решения позволило бы значительно упростить и сократить процесс разработки.

На сегодняшний день огромное количество программных продуктов строится по системе клиент-сервер. Такое решение предполагает наличие единого центра (сервера), который берёт на себя обработку поступающих команд в виде запросов и отправку ответа с результатом выполнения, и клиентов, задачи которых заключаются в формировании команд для сервера и наглядном представлении пользователю ответов с сервера.

Обычно при разработке сервера программисты создают интерфейс программирования приложения (API – Application Programming Interface), с помощью которого клиент может посылать строго определённые команды. Обычно API доступен только по протоколу HTTP или HTTPS. Это накладывает определённые требования к клиенту, ему необходимо поддерживать данный протокол. Для сложных систем это выливается в дополнительные финансовые затраты на приобретение дорогостоящего оборудования или в дополнительное время разработки программного продукта.

Решением данной проблемы может стать организация единого центра обработки удалённых команд, принимающего команды по различным протоколам и в различных форматах и приводящего их к единому формату, с которым и будут работать программисты. Разработчикам нужно будет выполнить лишь разовую настройку этого центра. Исходя из требований к разрабатываемой системе и из технических возможностей клиентов, можно определить список необходимых сетевых протоколов, которые требуется поддерживать, а также определить форматы запросов и ответов для каждого из них. Таким образом, единый центр обработки удалённых команд возьмёт на себя роль переводчика, который интерпретирует поступающие команды в понятный для сервера формат, и в обратном направлении переводит ответы сервера в понятный для клиента формат.

Главным достоинством данного решения является сокращение временных затрат при разработке серверного ПО. При проектировании многокомпонентных систем такое решение позволяет использовать устройства, работающие через разные сетевые протоколы, что может значительно сократить финансовые расходы. Использование низкоуровневых сетевых протоколов позволит увеличить их время автономной работы от батарей, т.к. существенно снизится объём передаваемых данных.

АРХИТЕКТУРА ОДНОСТРАНИЧНОГО ПРИЛОЖЕНИЯ ПО ОБРАБОТКЕ ДАННЫХ СЕРВИСА МИКРОБЛОГИНГА TWITTER

Белорусский государственный университет информатики и радиоэлектроники

г. Минск, Республика Беларусь

Череватенко Н. П.

Самаль Д. И. – канд. техн. наук, доцент

Первый веб-сайт появился двадцать лет назад и представлял из себя статическую страницу с текстом, спустя некоторое время компания Microsoft предоставила технологию ActiveX, появились каскадные таблицы стилей, веб страницы стали приобретать форму и функциональность. Они превратились в сложнейшие приложения по обмену информацией, просмотра медиа-файлов, место запуска игр и даже обработки графики. Изменились и методики их написания.