

Имитация периферийных устройств в системе дистанционного доступа к FPGA-комплексам

Ю. Е. Зинченко, В. С. Ленч, Т. А. Зинченко, В.Н. Павлыш
Донецкий национальный технический университет
zinchenko.tatyana@gmail.com

Ю. Е. Зинченко, В. С. Ленч, Т. А. Зинченко, В.Н. Павлыш Имитация периферийных устройств в системе дистанционного доступа к FPGA-комплексам. Целью работы является разработка и реализация метода имитации периферийных устройств, подключаемых к FPGA, а также разработка личного кабинета пользователя на Веб-сайте виртуальной лаборатории, и обеспечение возможности удаленного программирования FPGA-комплекса. Данная разработка необходима для упрощения доступа к дорогостоящим FPGA-комплексам студентам и инженерам, работающим в сфере компьютерной инженерии на базе FPGA. Объектом исследования являются методы имитации физических устройств, подключаемых к ПЛИС типа FPGA, которые позволяют организовать удаленный доступ к отладочным FPGA-комплексам. В работе был выполнен анализ существующих методов имитации различных физических устройств для обеспечения удаленной отладки FPGA-проектов. Были исследованы способы организации удаленного доступа к программируемым комплексам. На основании результатов анализа был разработан метод имитации периферийных устройств, подключаемых к FPGA-комплексу, с возможностью удаленного управления.

Ключевые слова: ПЛИС, FPGA, VHDL, РЭА, ЭВА, FPGA Remote Control, эмуляция, удаленный доступ, Internet

1 Введение

Сегодня в мире в целом и в странах СНГ в частности широко развиваются FPGA-технологии как одна из альтернатив проектирования аппаратных средств радиоэлектронной (РЭА) и электронно-вычислительной (ЭВА) аппаратуры. Основой этих технологий являются программируемые логические интегральные схемы (ПЛИС), и прежде всего их разновидность – FPGA (Field Programmable Gate Arrays), и языков описания (программирования) аппаратуры, таких как VHDL и Verilog[1]. Процесс проектирования на базе FPGA-технологий на этапе прототипирования обычно предусматривает использование FPGA-комплексов, однако это сдерживается относительной их дороговизной, широкой номенклатурой и необходимостью постоянно обновлять парк таких комплексов. Последнее особенно чувствительно для вузов и не только стран СНГ, но для западного региона, где доступ к ограниченному составу FPGA-комплексов нужно обеспечивать широкому кругу студентов и аспирантов. Поэтому разработка систем дистанционного доступа широкого круга специалистов и прежде всего студентов к FPGA-комплексам является, безусловно, задачей актуальной. Основной проблемой систем дистанционного

доступа к РЭА и ЭВА является обеспечение возможности имитации периферийных устройств, чтобы создавать эффект присутствия пользователя вблизи аппаратуры и как бы воспроизводить манипуляции органами управления. К настоящему времени известно уже множество таких систем, как к FPGA-комплексам, так и к другим электронным комплексам на базе микропроцессоров и микроконтроллеров[2-8]. В большинстве таких систем возможность имитации либо вообще не внедрена, либо предусматривает использование дополнительных программно-аппаратных средств, что резко усложняет, удорожает и снижает гибкость дистанционного доступа. В данной работе предлагается универсальный подход организации системы дистанционного доступа к FPGA-комплексам на базе сети Интернет с возможностью имитации периферийных устройств без использования дополнительного оборудования. Подход базируется на погружении специальных имитаторов в FPGA-кристалл в виде IP-ядер, разработанных на VHDL или Verilog, вместе с тестируемым проектом, которые способны воспринимать внешние манипуляционные команды от дистанционно удаленного оператора-отладчика проекта (инженера,

студента и т.п.). Статья является дальнейшим развитием работ [9-11], проводимых лабораторией ДонНТУ «FPGA-технологии проектирования и диагностика КС», в указанной области исследования [12].

1. Разработка метода имитации периферийных устройств для FPGA-комплекса

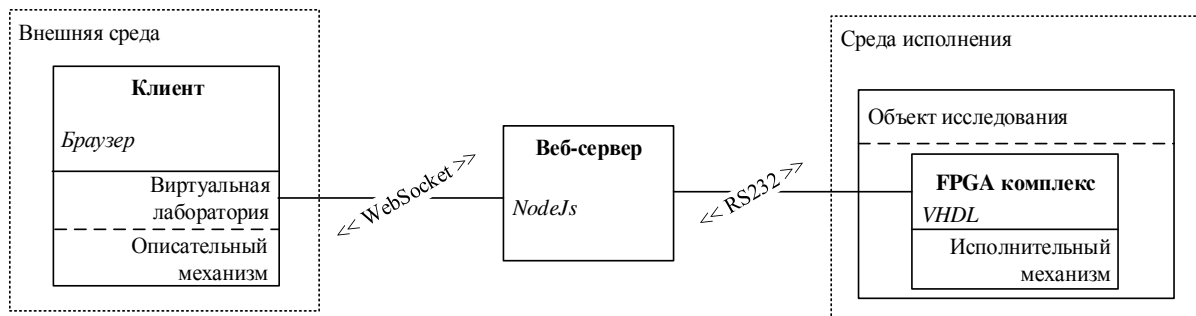


Рисунок 1 – Общая структура метода имитации в виртуальной FPGA-лаборатории

Как видно из структуры, клиентская часть находится во внешней среде, это означает, что она взаимодействует с пользователем посредством доступных устройств ввода-вывода. Также на схеме показано, что виртуальная лаборатория использует интернет-браузер для взаимодействия с веб-сервером. Такой выбор связан с отсутствием установки дополнительного программного обеспечения на персональный компьютер (ПК) пользователя, а интернет-браузер на сегодняшний день является неотъемлемой частью ПК, из-за большой популярности сети Интернет. Но такой выбор вносит свои ограничения на использование подключенного периферийного оборудования к ПК, так как это обусловлено политикой безопасности разработки интернет-браузеров.

Исходя из выбранного метода логика работы имитаторов описывается в клиентской части виртуальной лаборатории, это значит, что данные приходят от объекта исследования, а решение об использовании данных принимается на странице виртуальной лаборатории. Такой подход является менее безопасным с точки зрения использования данных, но позволяет снизить нагрузку и расход ресурсов FPGA-комплекса.

В качестве веб-сервера предпочтительно использовать серверное программное обеспечение, предназначенное для создания веб-приложений реального времени, как, например, Node.js. В сервере Node.js реализована поддержка

Выбранный метод, в связи с общей схемой реализации виртуальных лабораторий с дистанционным доступом, представляет собой общую структуру метода имитации периферийных устройств в виртуальной FPGA-лаборатории, представленной на рисунке 1.

протокола WebSocket, который позволяет передавать данные от сервера к клиенту без дополнительных запросов, что обеспечивает обмен данными в реальном времени. Из схемы на рисунке 1 видно, что серверная часть служит звеном взаимодействия между виртуальной лабораторией и FPGA-комплексом. Для связи FPGA-комплекса с сервером необходимо использовать какой-либо быстродействующий интерфейс передачи данных, как, например, RS232 или USB.

В данном случае объектом исследования и средой исполнения эмуляторов периферийных устройств является FPGA-комплекс. Также в нем содержится исполнительный механизм эмуляторов периферийных устройств, это означает, что в FPGA-комплексе расположена их аппаратная составляющая. Для того, чтобы пользователи могли разрабатывать свои проекты с использованием эмуляторов периферийных устройств, необходимо дополнять свои проекты вспомогательной заготовкой, где реализован исполнительный механизм эмуляторов устройств, а также обмен данными с сервером. Также возможно дополнять автоматически проект на этапе синтеза пользовательского устройства в FPGA, но для этого необходимо разработать дополнительное программное обеспечение для синтеза устройств.

Таким образом, структура проекта в FPGA должна иметь следующий вид, как показано на рисунке 2.

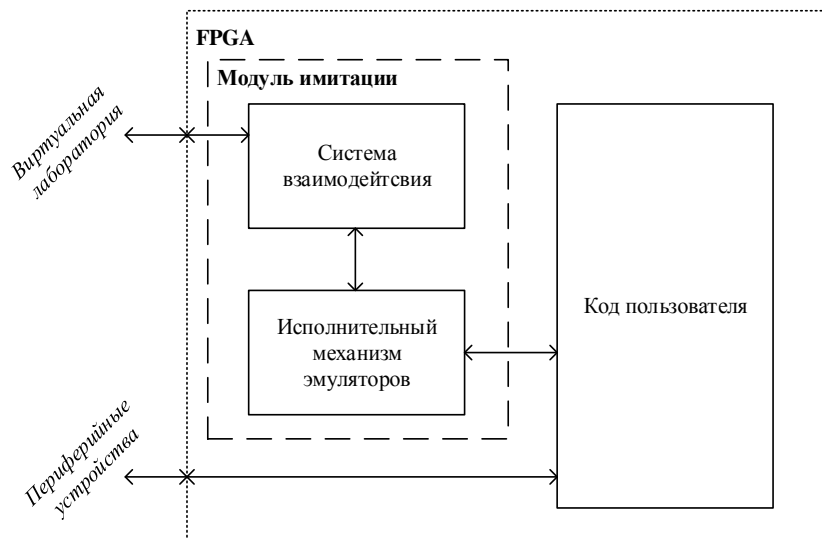


Рисунок 2 – Схема организации FPGA-проекта с эмуляторами периферийных устройств

На рисунке 2 видно, что код пользователя может использовать, как эмуляторы периферийных устройств, так и физически подключенные периферийные устройства. Такая организация связана с тем, чтобы разработчик имел возможность с минимальными трудозатратами изменять свой проект для использования, как в системе виртуальной лаборатории, так и при непосредственном доступе к периферийному оборудованию FPGA-комплекса.

Так как подключение к FPGA-комплексу организуется единым каналом связи, то необходимо разрабатывать систему взаимодействия сервера с имитаторами на плате. Задача системы взаимодействия принять данные с виртуальной лаборатории и перенаправить тому имитатору устройства, которому эти данные предназначены. Таким же образом обеспечить прием данных от имитаторов и передать их в виртуальную лабораторию.

Исполнительный механизм имитаторов представляет собой множество эмуляторов периферийных устройств, которые связаны с пользовательским кодом на уровне физических сигналов по протоколам связи, которые используют эти устройства. А с системой взаимодействия в том виде, который удобен для передачи по выбранному каналу данных, например, в виде цифровых байт данных.

Таким образом, выбранная методика имитации периферийных устройств в виртуальной FPGA-лаборатории с дистанционным доступом подразумевает под собой работу в несколько этапов. Имитатор ввода данных, в виртуальной лаборатории,

принимает данные от пользователя и обрабатывает их в соответствии с логикой работы устройства. Далее обработанные данные должны быть переданы на веб-сервер, который в свою очередь пересылает их в FPGA-комплекс по быстродействующему каналу связи. В FPGA данные должны быть приняты системой взаимодействия, которая идентифицирует целевой имитатор устройства, и пересылает ему данные. Имитатор устройства должен получить данные, преобразовать их в соответствии с протоколом работы соответствующего устройства, и переслать их в проект пользователя, который реализует поставленную задачу. В случае вывода данных необходимо произвести обратный процесс.

2. Разработка структуры взаимодействия виртуальной лаборатории с FPGA-комплексом

В общем случае виртуальная лаборатория обменивается данными с веб-сервером по сети Интернет, а сервер обменивается данными с FPGA-комплексом по физическому каналу связи. Для связи сервера с FPGA-комплексом предлагается использовать протокол передачи данных RS232, из-за простоты работы и широкой распространенности. Таким образом, следует учитывать, что протокол RS232 является последовательным, и использует две отдельные линии связи: для приема и для передачи. На рисунке 3 приведена общая схема системы взаимодействия между сервером и имитаторами в FPGA-проекте.

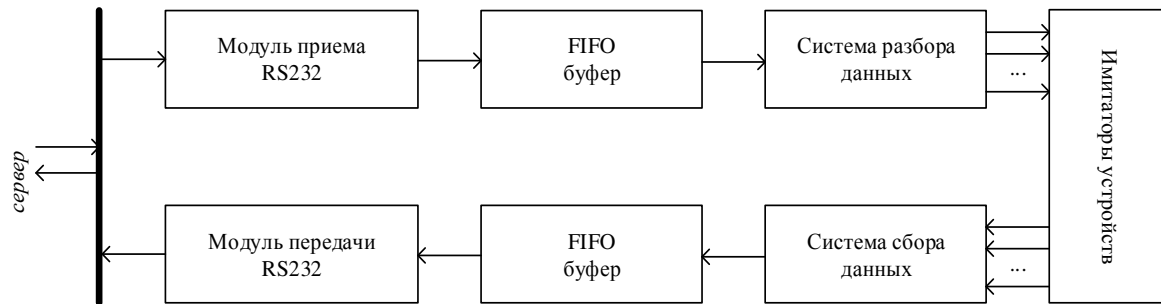


Рисунок 3 – Общая схема взаимодействия сервера с FPGA-комплексом

Такая схема позволяет с минимальными трудозатратами изменить способ обмена данными между сервером и FPGA-комплексом. Достаточно изменить модули передачи и приема в соответствии с выбранным интерфейсом связи, и схема продолжит свою работу в обычном режиме.

На схеме представлены модули приема и передачи RS232 они обеспечивают обмен данными по протоколу RS232 между FPGA-комплексом и сервером. Задачей этих модулей является обеспечение синхронизации передачи данных, а также преобразование данных из сигналов протокола связи в байты и наоборот.

Для увеличения стабильности обмена данными используются FIFO-буферы. При приеме данных может возникнуть ситуация, когда имитаторы еще не завершили прием предыдущей посылки, но сервер уже передает следующую порцию данных. В таком случае в

буфере будут накапливаться данные, и по мере работы имитаторов освобождаются. Таким же образом работает и буфер при передаче данных на сервер. Если от имитаторов устройств поступает много данных, но модуль передачи не успевает их переслать, тогда данные накапливаются в буфере и постепенно передаются на сервер.

Система разбора данных выполняет две функции: разбирает принятый пакет данных и передает данные имитатору устройства. Так как есть необходимость идентифицировать устройства, то необходимо разработать структуру пакета обмена данными. Эта структура должна содержать разделитель пакетов, идентификационный код устройства, данные и признак окончания пакета. Таким образом, была разработана структура пакета данных, представленная на рисунке 4.

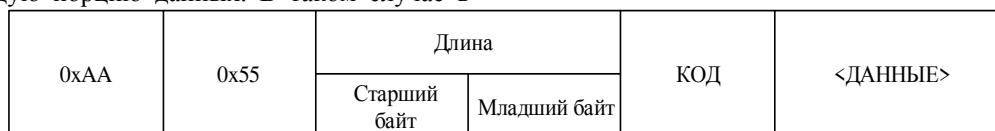


Рисунок 4 – Структура пакета данных между сервером и FPGA-комплексом

В представленной структуре двухбайтная подпись 0xAA55 является идентификатором начала пакета данных. Далее идут два байта длины данных, это поле определяет, сколько байт данных передается в пакете, и служит признаком окончания пакета. Поле «код» имеет длину один байт и представляет собой идентификационный код устройства. Поле «<данные>» содержит в себе количество байт, указанных в поле «длина». Такая структура пакета данных должна обеспечивать целостную передачу и однозначную пересылку данных между описательным и исполнительным механизмами эмуляторов периферийных устройств.

Таким же образом система сбора данных получает данные от имитаторов устройств, готовых к передаче, и формирует пакет для передачи на сервер.

Сервер в свою очередь должен обеспечивать прием и разбор пакетов данных от FPGA-комплекса и передачу их на веб-страницу виртуальной лаборатории. А также прием данных с веб-страницы виртуальной лаборатории, формирование пакета данных и передачу его в FPGA-комплекс.

На рисунке 5 приведена диаграмма программной организации имитаторов в виртуальной лаборатории.

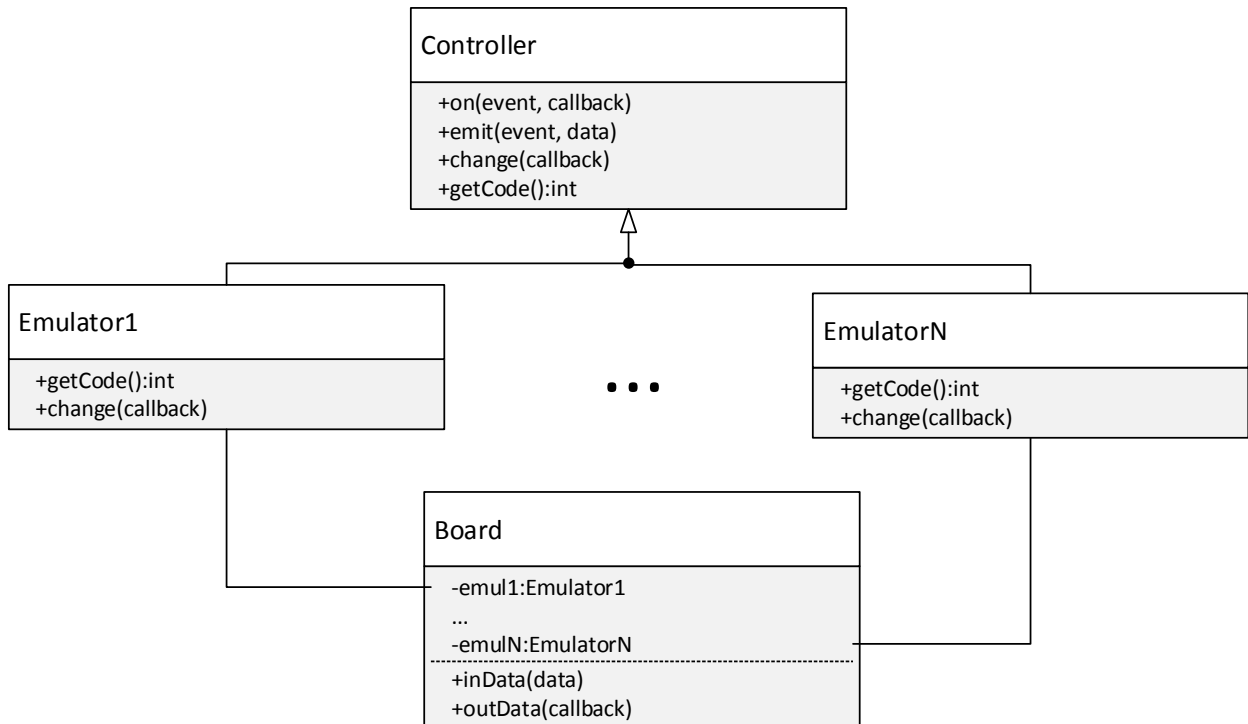


Рисунок 5 – Диаграмма программной организации имитаторов в виртуальной лаборатории

В общем случае каждый имитатор в виртуальной лаборатории должен расширять класс *Controller* и реализовывать методы *getCode()* и *change()*. Класс контроллер должен содержать в себе реализации методов *on()* и *emit()*, которые предназначены для организации событийной модели имитаторов. Метод *on(event, callback)* должен принимать на вход название события и функцию обработчик, которая будет вызываться при наступлении события, например, *on('change', function(data) { /* some actions */ })*. А метод *emit(event, data)* должен вызывать все обработчики, которые имеет событие *event*, и передавать им данные *data* в качестве параметра.

Метод *getCode()* должен возвращать целочисленный код имитатора устройства, который служит идентификатором связи с имитатором в FPGA-комплексе. Метод *change(callback)* должен представлять обертку над методом *on()* с типом события «change».

Экземпляры имитаторов должны находится в классе *Board*, который содержит два метода: *inData()* и *outData()*. Метод *inData(data)* должен вызываться главным приложением, которое принимает данные от сервера, и в качестве параметра ему передаются данные, которые содержат код, размер и сами данные. В зависимости от кода этот метод должен вызывать обработку соответствующего имитатора устройства. Метод *outData(callback)*

принимает на вход функцию обработчик, которая пересылается в метод *change()* всем экземплярам имитаторов. Обработчик, который передается в метод *outData()* должен содержать в себе обработку передачи данных на сервер. Данные должны передаваться в качестве параметра к функции обработчика, и должны иметь общую структуру для всех имитаторов устройств.

Каждый класс эмулятора должен работать с пользовательским интерфейсом веб-страницы виртуальной лаборатории и взаимодействовать с пользователем посредством различных элементов управления. Эмуляторы должны содержать в себе логику работы своего устройства и передавать данные методом *emit()*.

3. Структура систем разбора и сбора данных в FPGA-комплексе

Важнейшими частями в, разрабатываемой, системе имитации периферийных устройств являются системы разбора и сбора, пакетов данных в FPGA-комплексе. Эта система служит прикладным протоколом распознавания данных, принятых по физическому каналу связи.

На рисунке 6 приведена диаграмма состояний модуля разбора пакета данных.

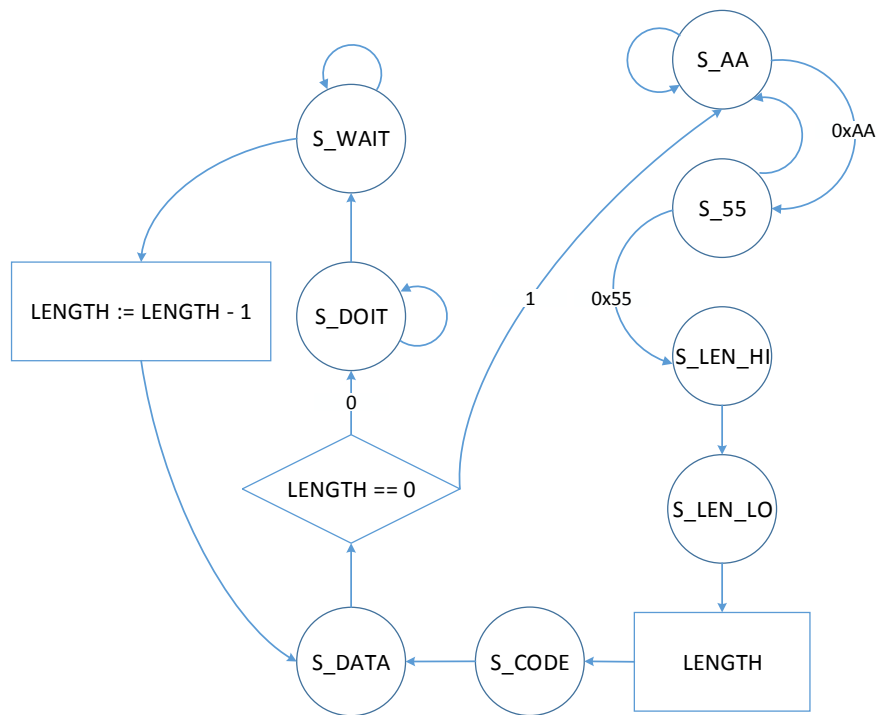


Рисунок 6 – Диаграмма состояний разбора пакета данных

Входным значением для системы разбора пакета данных является принятый байт. Первоначальным состоянием является состояние S_AA, это состояние, в котором ожидается первый байт стартовой сигнатуры пакета данных – 0xAA. Если приходит первый байт пакета данных, то состояние изменяется на S_55, которое ожидает второй байт сигнатуры – 0x55. Если двухбайтная сигнатура принята, то это означает, что далее идет корректная последовательность данных. Вначале принимается два байта длины данных, которые сохраняются в памяти. Далее принимается код имитатора устройства, который является адресом для целевого имитатора. После чего начинается пересылка данных имитатору устройства.

При пересылке данных имитатору проверяется значение длины данных. Если длина равняется нулю, то состояние переводится к начальному (S_AA). Если длина данных еще не обнулилась, то ожидается готовность приема данных имитатором устройства. Если имитатор готов к приему данных, то ему передается байт данных и состояние переводится в S_WAIT, где ожидается успешный прием имитатором устройства и после чего уменьшается на единицу длина данных. Далее состояние переводится в S_DATA, и эти операции повторяются до тех пор, пока имитатору не будут переданы все данные от сервера. Такая диаграмма разбора данных должна позволить реализовать систему

разбора пакета данных, как в FPGA-комплексе, так и на сервере.

Диаграмма состояний модуля сбора пакета данных от имитаторов периферийных устройств представлена на рисунке 7.

Схема сбора пакета данных похожа на схему разбора, только первым состоянием здесь является S_WAIT, в котором ожидается поступление данных от устройства. Когда на каком-либо имитаторе появляются данные, то перед переводом в состояние S_DOIT сохраняется его код устройства и длина данных. Состояние S_DOIT является переходным состоянием от состояния ожидания и начала передачи данных. После состояния S_DOIT каждое последующее состояние представляет собой пересылку байт, соответствующих структуре пакета данных. Так состояния S_LEN_HI и S_LEN_LO передают старшую и младшую части длины данных, а состояние S_CODE передает код имитатора устройства. Состояние S_DATA может быть расширено дополнительной обработкой, в зависимости от реализации имитатора периферийного устройства. После окончательной передачи пакета данных состояние, снова, должно установиться в S_WAIT.

Представленные на рисунках 6 и 7 диаграммы работы с пакетами данных позволяют реализовать обмен данными между зависимыми частями имитаторов устройств в виртуальной

лаборатории и FPGA-комплексе, который не зависит от физического канала связи.

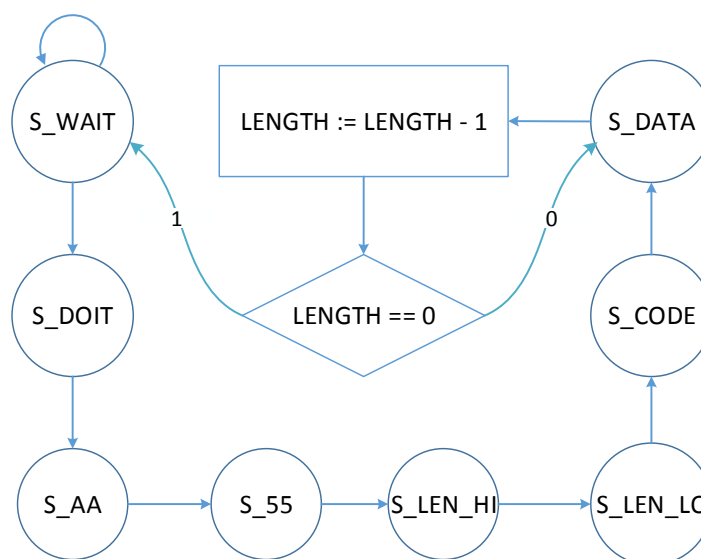


Рисунок 7 – Диаграмма состояний сбора пакета данных от имитаторов устройств

4. Общая структура исполнительного механизма имитатора устройства

Каких-либо жестких требований к реализации имитаторов периферийных устройств в FPGA нет. Но обязательно должны присутствовать, так или иначе, входные и выходные данные. На рисунке 8 представлена общая структура имитатора периферийного устройства. Пунктирной линией выделены входы и выходы, которые могут отсутствовать, в зависимости от реализуемого устройства.

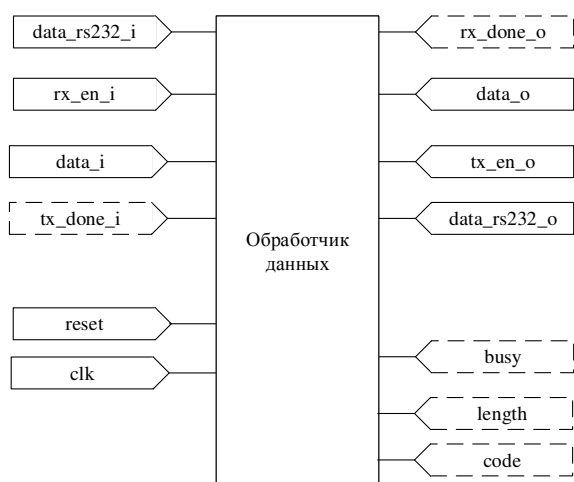


Рисунок 8 – Общая структура исполнительного механизма имитатора устройства в FPGA

Входы «*reset*» и «*clk*» являются обязательными для каждого устройства. Вход «*reset*» подключается к общей линии сброса, по которой происходит инициализация всей системы проекта. Линия «*clk*» является общей линией синхронизации, и несет в себе тактовый сигнал с частотой FPGA-комплекса. Выходы «*length*» и «*code*» должны присутствовать, так или иначе. Они не обязательно должны быть представлены, как выходы имитатора, а могут существовать, как константные значения и связаны с символьным наименованием имитатора устройства в системе. Значение «*code*» должно представлять собой идентификатор устройства, а значение «*length*» – количество выходных данных имитатора, передаваемых в виртуальную лабораторию.

Выход «*busy*» также не является обязательным, и предназначен для оповещения системы о том, что обработчик имитатора устройства, в данный момент, занят. Вход «*data_rs232_i*» предназначен для ввода данных, которые приходят от сервера, а вход «*rx_en_i*» должен оповещать имитатор о наличии новых данных на входе, готовых к приему. Об окончании приема порции данных может оповещать выход «*rx_done_o*». После обработки входных данных от лаборатории на выход «*data_o*» должны передаваться данные, в физических сигналах устройства, в пользовательский проект, где они должны использоваться в соответствии с заданной логикой.

На вход «*data_i*» поступают данные из пользовательского проекта в виде физических сигналов устройства. После их обработки на выход «*data_rs232_o*» передаются эти данные, готовые к передаче в виртуальную лабораторию. О наличии данных на выходе свидетельствует линия «*tx_en_o*». А линия «*tx_done_i*» оповещает имитатор устройства об успешной передаче на сервер.

Информационные сигналы по большей части служат средствами синхронизации при передаче данных между системами сбора и разбора данных и эмуляторами периферийных устройств. Такая схема позволит разработать исполнительный механизм имитатора периферийного устройства и связать его с описательным механизмом в виртуальной лаборатории через систему взаимодействия.

Заключение

В данной работе предложен подход имитации периферийных устройств в составе виртуальной FPGA-лаборатории с дистанционным доступом, не требующий дополнительных аппаратных комплексов, что выгодно отличает его от указанных аналогов. Разработаны структуры дистанционного взаимодействия виртуальной FPGA-лаборатории с FPGA-комплексом, а также структуры имитаторов для различных периферийных устройств, а также структуры имитаторов для различных периферийных устройств. На базе FPGA-лаборатории ДонНТУ [12] разработана экспериментальная система дистанционного доступа к FPGA-комплексам фирмы Xilinx. Проведенные эксперименты на FPGA-комплексе Spartan-3E показали положительные результаты. В настоящее время проводятся дальнейшие испытания разработанной системы.

Литература

1. Зинченко Ю., Калашников В., Хайдук С. и др. FPGA-технологии проектирования и диагностика компьютерных систем / Сб. научн. тр. VI Межд. научн.- практ. конф. «Современные информационные технологии и ИТ-образование». - Москва: МГУ, 2011. - Т. 1. 787 С. 422-429.
2. Fotopoulos V. Remote FPGA Laboratory Course Development based on an Open Multimodal Laboratory Facility / V. Fotopoulos, A. Fanariotis, T. Orphanoudakis, A. Skodras // [Электронный ресурс]: материалы 19-й всегреческой конф. по информатике (PCI 2015), Афины, Греция, 1-3 октября 2015. — Режим доступа: https://www.researchgate.net/publication/281437451_Remote_FPGA_Laboratory_Course_Development_based_on_an_Open_Multimodal_Laboratory_Facility

281437451 Remote FPGA Laboratory Course Development based on an Open Multimodal Laboratory Facility

3. Шауэрман А.А., Лаборатория Электронных Средств Обучения (ЛЭСО) СибГУТИ / А.А. Шауэрман, А.А. Подберезный [Электронный ресурс]. — Режим доступа: <http://www.labfor.ru/>

4. Reichenbach M. A New Virtual Hardware Laboratory for Remote FPGA Experiments on Real Hardware // M. Reichenbach, M. Schmidt, B. Pfundt, D. Fey. [Электронный ресурс]. — Режим доступа: <http://iuskm.donntu.org/electronic/iuskmm2017.pdf>

5. Soares J. A Remote FPGA Laboratory for Digital Design Students // Joao Soares, Jorge Lobo. ISR – Institute of Systems and Robotics DEEC, FCT, University of Coimbra [Электронный ресурс]. — Режим доступа: <http://ap.isr.uc.pt/archive/jlobo-REC2011.PDF>

6. Karthik S. Remote FIELD-Programmable Gate Array (FPGA) Lab // S. Karthik, P. Shreya, Srihari P, N.M. Viswanath. International Journal of Research in Engineering and Technology. Volume: 03 Issue: 04. Apr-2014.

7. Morgan F. Remote FPGA Lab With Interactive Control and Visualization Interface // F. Morgan, S. Cawley, M. Kane, A. Coffey, F. Callaly. Irish Signals & Systems Conference and China-Ireland International Conference on Information and Communications Technologies, Limerick, Ireland, pp. 221-226, Jun 2014 [Электронный ресурс]. — Режим доступа: <http://ieeexplore.ieee.org/document/6044871/>

8. Morgan F. Remote FPGA Lab for Enhancing Learning of Digital Systems / F. Morgan, S. Cawley, D. Newell. ACM Transactions on Reconfigurable Technology and Systems (TRETS), pp. 496-499, Oct 2012 [Электронный ресурс]. — Режим доступа: <http://ieeexplore.ieee.org/abstract/document/598152>

9. В.С. Ленч, В.А. Гень, Ю.Е. Зинченко, А.В. Коваль. Исследование организации удаленного доступа к FPGA-лаборатории // Компьютерная и программная инженерия – 2015 [Электронный ресурс]: материалы межд. научно-техн. конф. студентов, аспирантов и молодых ученых, г. Донецк, 15-16 декабря 2015г., с. 43. — Режим доступа: <http://cs.conf.donntu.org/docs/sbornik2015.pdf>

10. В.С. Ленч, Ю.Е. Зинченко, А.В. Коваль, Т.А. Зинченко. Исследование методов создания эмуляторов периферийных устройств для удаленной отладки проектов в FPGA-лаборатории // Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2016) [Электронный ресурс]: материалы I научно-

практ. конф., г. Донецк, 16-17 ноября 2016 г. с. 221. — Режим доступа: <http://www.pi.conf.donntu.org/ПЛИВС-2016%20сборник%20докладов%20студ%20секции.pdf>

11. А.Б. Азаров, Д.А. Зубков, В.С. Константинов, Ю.Е. Зинченко, А.В. Коваль. Эмуляция периферийных устройств для удаленного доступа к FPGA-комплексам // Информатика, управляющие системы, математическое и компьютерное моделирование в рамках III форума «Инновационные

перспективы Донбасса» (ИУСМКМ - 2017) [Электронный ресурс]: материалы VIII Межд. научно-техн. конф., г. Донецк, 26 мая 2016 г. с. 686. — Режим доступа: <http://iuskm.donntu.org/electronic/iuskmm2017.pdf>

12. Зинченко Ю.Е., Калашников В.И., Дяченко О.Н., и др.. Современные проекты FPGA-лаборатории ДонНТУ / Материалы межд. научно-практ. конф. «Инновационные перспективы Донбасса», том 5. «Компьютерные науки и технологии, г. Донецк, 20-22 мая 2015 г. – Донецк: ДонНТУ, 2015. С 60-66

Ю.Е. Зинченко, В.С. Ленч, Т.А. Зинченко, В.Н. Павлыш *Имитация периферийных устройств в системе дистанционного доступа к FPGA-комплексам. Целью работы является разработка и реализация метода имитации периферийных устройств, подключаемых к FPGA, а также разработка личного кабинета пользователя на Веб-сайте виртуальной лаборатории, и обеспечение возможности удаленного программирования FPGA-комплекса. Данная разработка необходима для упрощения доступа к дорогостоящим FPGA-комплексам студентам и инженерам, работающим в сфере компьютерной инженерии на базе FPGA. Объектом исследования являются методы имитации физических устройств, подключаемых к ПЛИС типа FPGA, которые позволяют организовать удаленный доступ к отладочным FPGA-комплексам. В работе был выполнен анализ существующих методов имитации различных физических устройств для обеспечения удаленной отладки FPGA-проектов. Были исследованы способы организации удаленного доступа к программируемым комплексам. На основании результатов анализа был разработан метод имитации периферийных устройств, подключаемых к FPGA-комплексу, с возможностью удаленного управления.*

Ключевые слова: ПЛИС, FPGA, VHDL, PЭА, ЭВА, FPGA Remote Control, эмуляция, удаленный доступ, Internet

Y. E. Zinchenko, V.S. Lench, T.A. Zinchenko, V.N. Pavlysh. *Simulation of peripheral devices in the system of remote access to FPGA-complexes. The aim of the work is to develop and implement a method for simulating peripheral devices connected to the FPGA, as well as developing a personal user cabinet on the Virtual Laboratory Web site, and providing the possibility of remote programming of the FPGA-complex. This development is necessary to simplify access to expensive FPGA-complexes for students and engineers working in the field of computer engineering based on FPGA. The object of the study are methods for simulating physical devices connected to FPGA type FPGAs that allow remote access to FPGA-based debugging systems. In work the analysis of existing methods of imitation of various physical devices for maintenance of remote debugging of FPGA-projects has been executed. The methods of remote access to programmable complexes were explored. Based on the results of the analysis, a method was developed for simulating peripheral devices connected to an FPGA-complex, with the possibility of remote control.*

Keywords: FPGA, FPGA, VHDL, REA, EVA, FPGA Remote Control, emulation, remote access, Internet

Статья поступила в редакцию 20.09.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко