

УДК 004.05 + 004.031.6

РАЗРАБОТКА СТРУКТУРЫ СИСТЕМЫ ВЕРИФИКАЦИИ FPGA-ПРОЕКТОВ**Шерекин А.А., Зинченко Ю.Е., Зинченко Т.А., Корченко А.А.**

Донецкий национальный технический университет

кафедра компьютерной инженерии

E-mail: alexandr.sherekin@gmail.com**Аннотация**

Шерекин А.А., Зинченко Ю.Е., Зинченко Т.А., Корченко А.А. Разработка структуры системы верификации FPGA-проектов. Рассмотрена проблематика верификации и ее место в процессе проектирования аппаратных проектов на базе FPGA. Рассмотрены структурные элементы, используемые в процессе верификации. Разработан вариант структуры, согласно которой можно выполнять верификацию. Проанализированы положительные и отрицательные характеристики предлагаемой структуры.

Общая постановка проблемы

Процесс верификации представляет собой один из важнейших этапов процесса проектирования. Независимо от того, насколько велик проект, корректность его работы является одним из главных критериев качества. По причине того, что любой процесс проектирования зачастую сопровождается ошибками, верификация берет на себя ответственность за их нахождение. Вопрос верификации возникает не только при создании нового устройства или проекта, а также при портировании готовых проектов на другую элементную базу, как например, система, работающая на совокупности микропрограммируемых устройств, может быть перенесена на ПЛИС. Из-за различий аппаратной платформы могут возникать ошибки, которые сложно заранее предугадать. В таких ситуациях процесс верификации позволяет выяснить степень подобия нового проекта исходному и обнаружить как можно большее количество расхождений.

Исследования

Если взглянуть на процесс верификации в общем, то можно выделить несколько важнейших элементов структуры системы верификации:

верифицируемый объект – объект, который будет подвергаться различным манипуляциям и проверке на корректность работы. Если взять в качестве примера ситуацию, приведенную выше, то верифицируемым объектом будет новый проект, реализованный на ПЛИС.

эталонный объект – объект, который предоставляет корректные данные при воздействии на него входными сигналами, с которым будет выполняться сравнение результатов работы верифицируемого объекта. В качестве эталонного объекта могут выступать уже готовые аппаратные системы, если выполняется портирование проекта на другую аппаратную платформу. Если же исходный проект не доступен, либо его вовсе нет, тогда в качестве эталонного может выступать модель, реализованная, например, на языке описания аппаратуры VHDL. Прежде, чем использовать такую модель необходимо её тщательно промоделировать в различных САПР и убедиться в корректности работы.

генератор тестов – устройство, или программный модуль, выполняющий генерацию входных воздействий для передачи их на верифицируемый и эталонный объекты. Качество тестов позволяет покрывать то или иное количество возможных ошибочных ситуаций и обнаруживать их.

средство сравнения – устройство либо программный модуль, выполняющий непосредственное сравнение двух последовательностей битов и выдающий результат, равны

ли входные потоки данных либо нет.

метод верификации – способ, или методика, позволяющая делать вывод при появлении ошибок в работе верифицируемого объекта. В зависимости от того, насколько быстро реализуется поиск ошибки в проекте и реализуется ли вообще, зависит качество метода.

Разработка структуры системы верификации

В качестве одного из возможных вариантов структуры системы верификации предлагается реализация, приведенная на рис. 1

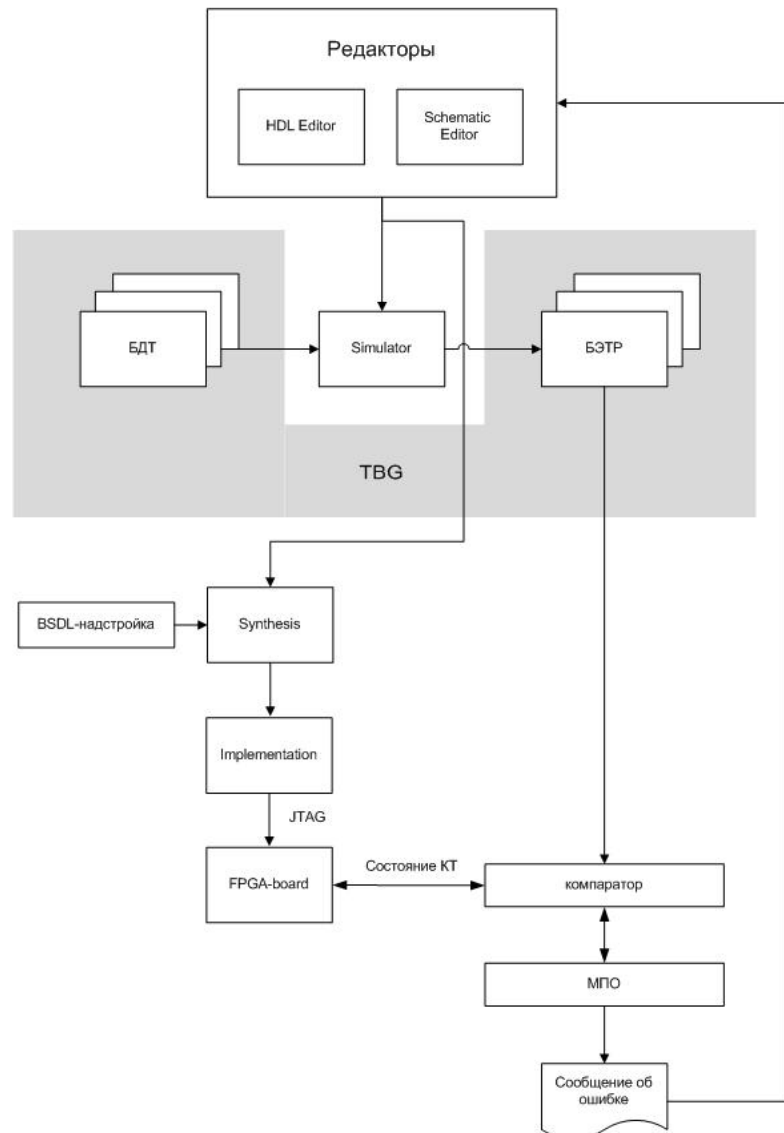


Рисунок 1 – Структура системы верификации

Данную структуру следует разделить на несколько основных частей и этапов для выполнения верификации:

выбор HDL-редактора. Так как в качестве эталонного объекта предлагается использовать модель проекта на одном из языков описания аппаратуры, то первым элементом структуры является HDL-редактор, с помощью которого создается эталон.

создание базы данных тестов. В зависимости от сложности верифицируемого объекта базу данных тестов можно реализовать вручную, либо использовать различные генераторы псевдослучайных тестов. Качество этого модуля во многом предопределяет

количество найденных ошибок.

симуляция эталонного объекта. После моделирования и отладки эталона на него необходимо подать сгенерированные входные воздействия. Результатом этого процесса будет являться

база эталонных тестовых реакций. С ней будет выполняться сравнение результатов работы верифицируемого объекта. Симулятор, база данных тестов и база эталонных тестовых реакций объединяются в один блок TBG – Test Bench Generator.

BSDL-надстройка. По причине того, что в разрабатываемой структуре в качестве средства верификации предлагается стандарт JTAG Boundary Scan, в ПЛИС (в данном случае предполагается FPGA от Xilinx), кроме основного проекта, зашивается еще и дополнительная логика, позволяющая получать доступ ко внутренним сигналам проектируемого устройства. Это позволяет отслеживать значения сигналов внутри устройства и дает возможность локализовать источник неисправности. BSDL-надстройка представляет собой IP-ядро ICON - встроенного контроллера, реализующего связь между компьютером проектировщика и интерфейсом JTAG, а также IP-ядро логического анализатора (ILA, VIO), который позволяет выводить внутренние сигналы верифицируемого устройства во внешнюю среду для анализа. Выбор логического анализатора зависит от количества сигналов, которые необходимо анализировать, и от размера устройства. VIO-анализатор предназначен для небольших схем, с количеством проверяемых сигналов до 256 [1]. ILA-анализатор позволяет анализировать большее количество сигналов, предоставляя для этих целей 16 портов по 256 сигналов в каждом [2]. Оба этих ядра могут настраиваться в зависимости от требований проекта и включаться в прошивку ПЛИС.

прошивка FPGA. После выполнения синтеза объединенных BSDL-надстройки и верифицируемого устройства, проект прошивается на ПЛИС.

подача входных воздействий и анализ результатов тестов. Предложенная схема верификации предполагает, что процесс сравнения и анализа выполняется на ПК проектировщика, где находится сам компаратор, а также модуль поиска ошибок (МПО). При этом анализируется результат сравнения и в случае отличия его от эталона выводится сообщение, о том, в каком именно блоке была обнаружена ошибка, либо, если это в текущий момент сделать невозможно, подаются дополнительные тестовые последовательности, с целью локализовать ошибку.

Из недостатков данного метода стоит выделить большие объемы информации, который необходимо передавать между ПЛИС и ПК для получения результатов тестовых воздействий, что особо критично при работе с большими проектами.

Однако с другой стороны, данный метод позволяет уменьшить аппаратные затраты из-за вынесения блока МПО из структуры ПЛИС, легко изменять алгоритмы, используемые для верификации проекта.

Выводы

Результатом данной статьи является структура системы верификации проектов на FPGA. Из-за ориентации на использование Boundary Scan и ПЛИС фирмы Xilinx предложенная схема имеет несколько особенностей, таких как использование готовых IP-ядер, позволяющих отслеживать процессы, происходящие внутри ПЛИС на экране монитора ПК и автоматизировать процесс обнаружения ошибок в проектах. Опыт разработки подобных схем будет использован при написании магистерской работы.

Список литературы

1. LogiCORE IP ChipScope Pro Virtual Input/Output (VIO) (1.04a) June 22, 2011
2. LogiCORE IP ChipScope Pro Integrated Logic Analyzer (ILA) (v1.04a) June 22, 2011
3. Сайт производителя FPGA, компании Xilinx [электронный ресурс]. - Режим доступа: <http://xilinx.com>