

ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ԿՐԹՈՒԹՅԱՆ ԵՎ ԳԻՏՈՒԹՅԱՆ  
ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

## Կրրիկյան Հարություն Ռուբենի

ՎԵՐԱԿԱՌՈՒՑԱԿՐԿՈՂ ՀԱՇՎՈՂԱԿԱՆ ՀԱՄԱԿԱՐԳԵՐԻ ԿԱՌՈՒՑՄԱՆ  
ՄԻՋՈՑՆԵՐԻ ՄՇԱԿՈՒՄ ԵՎ ՀԵՏԱԶՈՏՈՒՄ

### ՍԵՂՄԱԳԻՐ

Ե.27.01 «Էլեկտրոնիկա, միկրո և նանոէլեկտրոնիկա» մասնագիտությամբ  
տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի հայցման  
ատենախոսության

Երևան 2016

---

---

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ АРМЕНИЯ  
НАЦИОНАЛЬНЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ АРМЕНИИ

**Кррикян Арутюн Рубенович**

**РАЗРАБОТКА И ИССЛЕДОВАНИЕ МЕТОДОВ ПОСТРОЕНИЯ  
РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**

### АВТОРЕФЕРАТ

диссертации на соискание ученой степени кандидата технических  
наук по специальности 05.27.01–  
“Электроника, микро- и нанoeлектроника”

Ереван 2016

---

---

Ատենախոսության թեման հաստատվել է Երևանի պետական համալսարանում

Գիտական ղեկավար՝ տ.գ.դ. Վ.Շ. Մելիքյան

Պաշտոնական ընդդիմախոսներ՝ տ.գ.դ. Ս.Խ. Խուդավերդյան  
տ.գ.թ. Խ.Գ. Շարոյան

Առաջատար կազմակերպություն՝ ՀՀ ԳԱԱ Ռադիոֆիզիկայի և էլեկտրոնիկայի ինստիտուտ

Ատենախոսության պաշտպանությունը տեղի կունենա 2016թ. հունիսի 8-ին, ժամը 14<sup>00</sup>-ին, ՀԱՊՀ-ում գործող «Կառավարման, ավտոմատացման և էլեկտրոնիկայի» 032 Մասնագիտական խորհրդի նիստում (հասցեն՝ 0009, Երևան, Տերյան փ., 105, 17 մասնաշենք):

Ատենախոսությանը կարելի է ծանոթանալ ՀԱՊՀ-ի գրադարանում:  
Սեղմագիրն առաքված է 2016 թ. մայիսի 6-ին:

032 Մասնագիտական խորհրդի  
գիտական քարտուղար, տ.գ.դ.



Ա.Գ. Ավետիսյան

---

---

Тема диссертации утверждена в Ереванском государственном университете.

Научный руководитель: д.т.н. В.Ш. Меликян

Официальные оппоненты: д.т.н. С.Х. Худавердян  
к.т.н. Х.Г. Шароян

Ведущая организация: Институт радиофизики и  
электроники НАН РА

Защита диссертации состоится 8-го июня 2016г. в 14<sup>00</sup> ч. на заседании Специализированного совета 032 - “Управления, автоматизации и электроники”, действующего при Национальном политехническом университете Армении (НПУА), по адресу: 0009, г. Ереван, ул. Теряна, 105, корпус 17.

С диссертацией можно ознакомиться в библиотеке НПУА.

Автореферат разослан 6-го мая 2016г.

Ученый секретарь  
Специализированного совета 032  
д.т.н.



А.Г. Аветисян

## ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

**Актуальность темы.** Развитие цифровых интегральных схем (ИС) привело к широкому распространению проектов с многочисленными блоками интеллектуальной собственности, большим количеством процессоров и множеством областей тактовой частоты. Индустрия электронных схем движется в направлении роста проектов типа SoC. Это способствует значительному увеличению времени тестирования ИС (в 2014 году существенно увеличилось число проектов, в которых 80% времени разработки тратится на тестирование). Контроль за временем тестирования и количеством инженеров, занимающихся тестированием, - важная проблема для современных проектов цифровых ИС. Скорость увеличения эффективности тестирования не соответствует скорости увеличения количества элементов в ИС. Кроме того, для современных цифровых проектов время разработки программного обеспечения почти соразмерно времени разработки аппаратной части ИС. В случае традиционного последовательного метода проектирования (когда разработка программного обеспечения начинается после первой доступной интегральной схемы) можно не успеть закончить проект в запланированные сроки из-за задержек в окончании программной части. Исходя из вышеизложенного, актуальной проблемой является разработка новых методов увеличения эффективности и качества тестирования.

Использование прототипов на основе реконфигурируемых вычислительных (РВ) систем может значительно увеличить качество и эффективность тестирования ИС. Можно также начать разработку программной части, когда первый функциональный прототип доступен. Развитие программируемых логических интегральных схем (ПЛИС) дало возможность создания еще больших ИС в одном устройстве (4 миллиона эквивалентных ИС элементов в одной ПЛИС). Кроме того, возможность частичного программирования современных ПЛИС позволяет создавать новые типы систем тестирования.

Одна из основных проблем тестирования с помощью прототипа ПЛИС - это ограниченная возможность обнаружения причин провалов. Этот процесс сравним по сложности с обнаружением провалов в полупроводнике из-за ограниченной видимости внутренних сигналов прототипа. Чтобы снизить время тестирования, нужно улучшить этот процесс. Существующие методы разработки РВ систем не используют в полной мере возможности современных ПЛИС устройств для увеличения видимости. Они позволяют измерять сигналы только в небольшом промежутке времени. Для решения этой проблемы необходимо создать РВ системы, которые, используя синтезируемые утверждения, могли бы непрерывно измерять сигналы во время работы прототипа.

Отметим, что не существует метода оценки качества тестирования на прототипах ПЛИС. Синтезируемые утверждения можно использовать и для оценки функционального охвата.

Кроме вышесказанного, РВ системы можно использовать для исчерпывающего тестирования цифровых ИС с несколькими областями асинхронного тактового сигнала. Рассматривая работу прототипа, можно тестировать способы синхронизации сигналов, проходящих через разные области тактового сигнала. Но в случае провалов практически невозможно найти причину их появления. Чтобы решить эту проблему, нужно интегрировать синтезируемые утверждения в каждый

блок синхронизации для измерения сигналов и в случае неправильного поведения найти блок синхронизации.

**Объект исследования.** Методы нахождения функциональных провалов и вычисления функционального охвата в прототипах на основе РВ систем. Влияние этих методов на количество дополнительных ресурсов и на скорость прототипа.

**Целью диссертационной работы является:** разработка методов нахождения функциональных провалов в РВ системах, а также методов вычисления функционального охвата для тестирования в прототипе; разработка методов построения РВ систем для тестирования цифровых частей ИС; построение РВ систем для тестирования ИС с несколькими областями тактовой частоты.

**Методы исследования.** В процессе исследования были использованы методы моделирования электронных схем, автоматизированной разработки систем и средства описания на языках логических уровней и сред моделирования.

**Научная новизна работы.**

1. Разработана новая процедура построения цифровой части контроллера USB интерфейса с использованием синтезируемых утверждений, которая дает возможность увеличения видимости внутренних сигналов РВ системы.
2. Предложен метод оценки качества функционального тестирования с помощью внедренных утверждений.
3. Предложен метод построения РВ системы для тестирования уровня связи USB интерфейса с целью решения основных проблем функционального тестирования.
4. Разработан метод тестирования цифровых схем с несколькими областями тактовой частоты, в котором синтезируемые утверждения внедрены в каждый блок синхронизаций сигналов, проходящих через разные области тактового сигнала для обнаружения неправильного поведения синхронизаторов.
5. Предложен метод использования частичного перепрограммирования ПЛИС для снижения количества дополнительных ресурсов и увеличения скорости прототипа.

**Практическая ценность работы.** Разработанные методы построения РВ систем были исполнены в программной аппаратной среде TAID. TAID Instrumentor разработан для выполнения процедуры построения прототипа ПЛИС с внедренными утверждениями и удобным графическим интерфейсом. Предложенные методы значительно упрощают процесс нахождения причин ошибок и дают возможность вычисления функционального охвата с использованием 15...16% дополнительных ПЛИС ресурсов и незначительным снижением скорости прототипа. TAID Instrumentor дает возможность снизить количество дополнительных ресурсов среды тестирования с помощью частичного перепрограммирования ПЛИС и в итоге увеличить скорость прототипа.

**Достоверность научных положений** подтверждена моделированием приведенных научных результатов и сопоставлением результатов практических испытаний.

**Внедрение.** Программный и аппаратный инструмент TAID внедрен в ЗАО “Синописис Армения”. TAID Instrumentor используется для построения ПЛИС прототипов контроллеров USB3.0, USB2.0 и USB1.1 с внедренными синтезируемыми утверждениями. TAID Debugger используется для наблюдения статуса и контроля внедренной аппаратной части среды тестирования во время применения прототипа.

### **Основные положения, выносимые на защиту:**

1. Метод увеличения видимости внутренних сигналов и вычисления функционального охвата РВ системы с помощью внедренных синтезируемых утверждений.
2. Метод построения прототипа уровня связи USB интерфейса.
3. Метод тестирования цифровых схем с несколькими областями тактового сигнала с использованием РВ систем.
4. Программная аппаратная среда для построения прототипа и внедрения синтезируемых утверждений.

**Апробация работы.** Основные научные и практические результаты диссертации докладывались на:

- Международной конференции "CSIT: IEEE Computer Science and Information Technologies" (Ереван, Армения, 2015г.);
- 10-й Международной конференции "Semiconductor micro-and nanoelectronics" (Ереван, Армения, 2015г.);
- научных семинарах ЗАО "Синописис Армения" (Ереван, Армения, 2014-2015 гг.).

**Публикации.** Основные положения диссертации представлены в пяти научных публикациях, список которых приводится в конце автореферата.

**Структура и объем работы.** Диссертация состоит из введения, трех глав, основных выводов, списка литературы из 105 наименований и трех приложений. Основной объем диссертации составляет 121 страницу, включая 52 рисунка и 9 таблиц. Общий объем работы вместе с приложениями – 142 страницы. Диссертация написана на армянском языке.

## **ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ**

**Во введении** обоснована актуальность темы, сформулированы цель и основные задачи исследования, изложены изучаемые объекты и модели, представлены научная новизна, практическое значение работы и основные научные положения, выносимые на защиту.

**В первой главе** обоснована необходимость разработки средств построения РВ систем для тестирования современных ИС. Показаны основные функции и места применения прототипов. Представлены основные задачи и проблемы тестирования ИС с помощью РВ систем. Показаны современные методы построения РВ систем тестирования.

Система, в которой программируемая логика используется для вычисления, называется реконфигурируемой вычислительной системой. Такие системы широко используются для построения прототипов ИС. РВ системы стали доступны с 80-ых годов с появлением ПЛИС. Структура ПЛИС, которая позволяет практически неограниченное количество раз изменять функциональность, стала причиной появления новой дисциплины, где многие аппаратные алгоритмы исполняются в одном устройстве, как программные алгоритмы в процессоре. Большая скорость аппаратного исполнения алгоритма по сравнению с программной сделала этот метод привлекательным для решения многих проблем в разных сферах. Скорость систем, выполненных с помощью ПЛИС, высока для решения некоторых проблем, поскольку она может адаптироваться к алгоритму.

Для вычисления с помощью РВ систем используется минимальное количество логики для контроля. Разница между вычислениями РВ системы и процессора представлена на рис.1. В РВ системе используется множество единиц вычисления. Каждая единица состоит из контроллера интерфейса внешней памяти, регистров, сумматоров и умножителей. РВ система выполняет вычисления параллельно в сравнении с процессором, в котором операции проводятся последовательно.

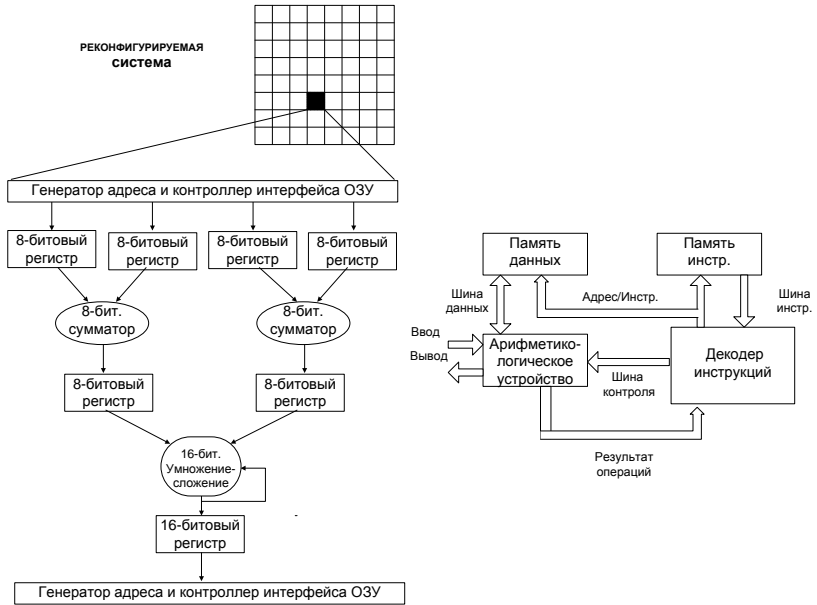


Рис. 1. Разница вычисления РВ систем и процессоров

Современные РВ системы в основном строятся на основе ПЛИС. Увеличение количества элементов по закону Мура является основной причиной использования этих устройств. Другие структуры реконфигурируемых устройств не нашли широкого применения.

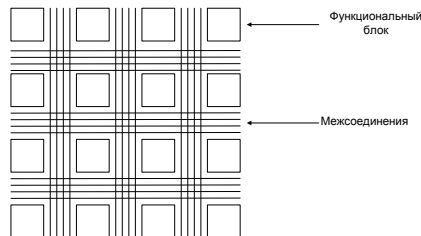


Рис. 2. Структура современных ПЛИС

ПЛИС состоит из массива функциональных блоков (ФБ), окруженных сетью межсоединений (рис. 2). Для основных логических операций используется таблица поиска, функция которой программируется на статическом оперативном запоминающем устройстве (ОЗУ) (рис.3а). ПЛИС содержит триггеры для осуществления последовательных операций. Простейший функциональный блок состоит из таблицы поиска, триггера и мультиплексора (рис.3б). Функциональный блок можно перепрограммировать, изменяя содержание таблицы поиска, контролирующей сигнал мультиплексора и начальное значение триггера. Каждая точка программирования контролируется одним битом статического ОЗУ.

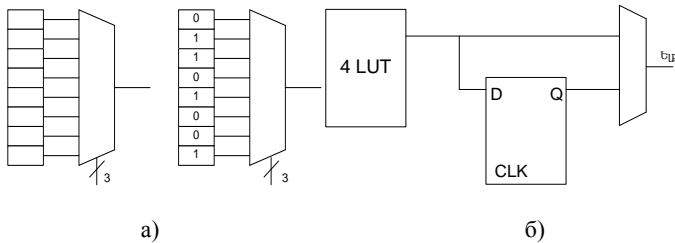


Рис. 3. Таблица поиска (а) и простейший функциональный блок (б)

Как уже было отмечено, ФБ окружены сетью межсоединений. Функциональные блоки используют программируемые межсоединения для доступа к этой сети, которые тоже используют биты статического ОЗУ для построения структуры схемы (рис. 4).

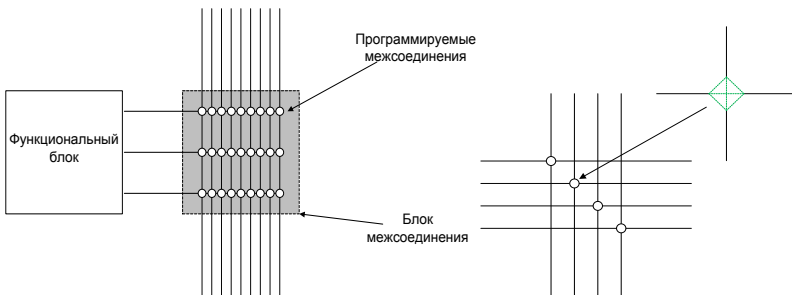


Рис. 4. Структура блока межсоединений

Применяется несколько методов построения межсоединений. Простейшая форма предназначена для соединения соседних ФБ.

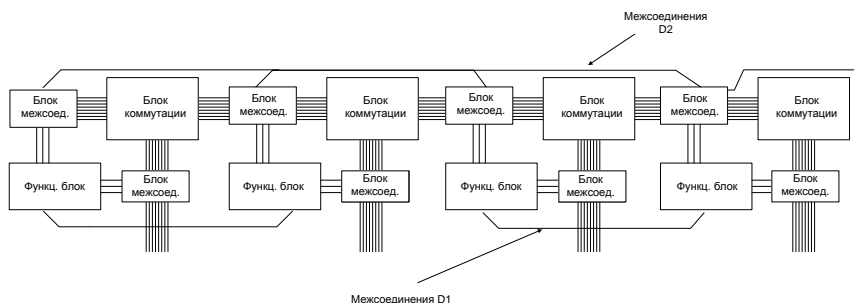


Рис. 5. Методы построения межсоединений

Этот метод неприменим для построения соединения ФБ на больших расстояниях, поскольку задержка сигнала растет линейно с увеличением дистанций, и, кроме того, ФБ, которые находятся на пути контакта, неприменимы для других целей. Чтобы решить эту задачу, используются типы межсоединений D2 и D3, которые обходят 2 и 3 ФБ соответственно почти с той же задержкой (рис. 5). Многие устройства применяют иерархическую структуру, в которой разные методы межсоединения используются на разных уровнях. На низшем уровне (1x1-4 ФБ) используется простейший метод, и длина межсоединения увеличивается с ростом уровня. В случае качественного синтеза простейшие межсоединения преобладают.

Изучение тенденций развития проектов цифровых ИС показало, что индустрия движется в направлении роста количества элементов, использования проектов типа SoC, увеличения количества процессоров и блоков интеллектуальной собственности, увеличения количества области тактового сигнала. В результате увеличивается время тестирования, что и приводит к задержкам окончания проекта. Необходимы новые методы, чтобы увеличить эффективность тестирования.

Большое количество времени тратится на функциональное тестирование (в среднем 57%). Это связано со следующими факторами:

- большое количество состояний, в которых система может находиться;
- нахождение неправильной функциональности;
- не существует устройства, функциональность которого можно считать абсолютно правильной, и сравнивать с ним разрабатываемое устройство.

Первый из факторов является важной проблемой, потому что функциональность есть в каждом состоянии, и все возможные переходы с одного состояния к другому должны тестироваться. Вторая проблема решена для моделирования с помощью SystemVerilog утверждений, мониторов и множества методов моделирования (UVM, VVM и т.д.). Решение третьей проблемы возможно только частично, так как модели устройств, которые используются во время моделирования, не могут полностью соответствовать спецификации.

Применение прототипов на основе РВ систем может улучшить эффективность и качество функционального тестирования. Прототип применяется по следующим причинам:

- с помощью прототипа можно преодолеть границы программного моделирования ИС, которое, начиная с некоторого количества элементов, невозможно эффективно использовать на недостаточной скорости и



производительности. Если точных моделей устройств не существует, то применение прототипа неизбежно;

- прототип предоставляет полностью функционирующие устройства для производства ИС. ПЛИС прототип позволяет начать разработки программной части до производства ИС, что значительно уменьшит задержки и позволит закончить проект вовремя;
- прототип является идеальной платформой для тестирования блоков интеллектуальной собственности и их взаимодействия с другими ИС.

Для построения и эффективного использования прототипа должны быть решены следующие проблемы:

1. Нужно разделить большие проекты на несколько ПЛИС устройств. Это важная проблема, поскольку возможность ошибок во время разделения велика. Кроме того, нужно большое количество межсоединений частей проекта в разных ПЛИС и использование множества выводов. Эта проблема может быть решена с помощью дополнительных логических схем, которые осложняют нахождение ошибок.
2. Много времени тратится на построение полностью функционирующего прототипа из-за ошибок. В случае отсутствия нужных устройств процесс нахождения ошибок может занять очень много времени. Причиной ошибок могут быть:
  - преобразование проекта для исполнения в ПЛИС технологии;
  - ошибки проектирования платы;
  - ошибки цифрового контроллера.
3. Сложность нахождения причин ошибок очень большая из-за ограниченной видимости внутренних сигналов ПЛИС. Инструментация современными методами лишь частично решает эту проблему. В этом случае следует иметь в виду:
  - инструментация возможна только для одной ПЛИС;
  - длительность наблюдения за сигналами возможна только на ограниченный период времени, так как для хранения информации используются блоки ОЗУ ПЛИС, и количество этих блоков ограничено;
  - чтобы инструментировать новые сигналы, нужно пройти весь процесс построения прототипа;
  - скорость прототипа уменьшается из-за использования дополнительных ресурсов ПЛИС.

Существует два метода наблюдения за сигналами, которые широко используются в индустрии. Для обоих методов вышеназванные ограничения уменьшают эффективность нахождения причин ошибок.

**Во второй главе** разработаны и изучены методы построения прототипов USB контроллера на основе РВ систем. Разработаны методы, способствующие нахождению ошибок в прототипе. Предложен метод вычисления функционального охвата и тестирования цифровых схем с несколькими тактовыми сигналами.

Утверждения можно представить как описание предназначенного поведения. Утверждения гарантируют соответствия между запланированной и реализованной логическими схемами.

```

propertyport_rst;
@(posedgeclk)
(reset_request) ##[1:3] ($past(fsm_prtsm) !=
PRT_RESET) && (fsm_prtsm ==
PRT_RESET) ##1 prt_dbc_pr;
endproperty
svc_u31prt_dbc_port_reset: assert property
(port_rst);

```

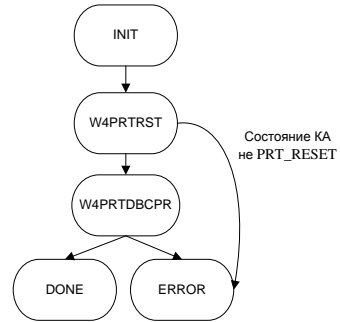


Рис. 6. Пример SVA кода и синтезируемого конечного автомата

Пример утверждения на языке SVA и синтезируемого конечного автомата (КА) представлен на рис. 6. В случае сигнала сброса этим утверждением проверяется следующее поведение:

- сигнал сброса порта становится активным;
- в течение трех периодов тактового сигнала состояние конечного автомата порта должно быть PRT\_RESET;
- и, наконец, после одного периода сигнала сброса, предназначенный для тестовой части (prt\_dbc\_pr) порта, должен стать активным.

Утверждения способствуют нахождению провалов следующим образом:

- позволяют более точно описать поведение, которое может длиться несколько периодов;
- предоставляют возможность более точно описать предположения, сделанные во время проектирования;
- увеличивают видимость внутренних сигналов прототипа;
- являются методом вычисления функционального охвата.

Чтобы способствовать нахождению ошибок, предложена новая процедура построения цифровой части контроллера USB с использованием синтезируемых утверждений (рис. 7). Предложено внедрение утверждения в уровень регистровых передач (УРП) контроллера USB во время разработки.

Синтезируемые УРП утверждений и контроллера USB разрабатываются одновременно и тестируются в среде проверки контроллера. Синтезируемые утверждения позволяют уменьшить время нахождения причин ошибок. Часть тестов моделирования можно проводить с помощью прототипа, что значительно уменьшает время разработки контроллера.

Уровень связи USB3.0 является важнейшей частью контроллера, поскольку основные ограничения скорости возникают из-за ошибок во время разработки этого блока. Полное соответствие спецификаций этого блока обязательно, так как это позволяет соединять другие части контроллера с внешним миром. Широко распространены ошибки протокола интерфейса уровней связи с аналоговой частью, и нахождение этих причин - очень сложная проблема.

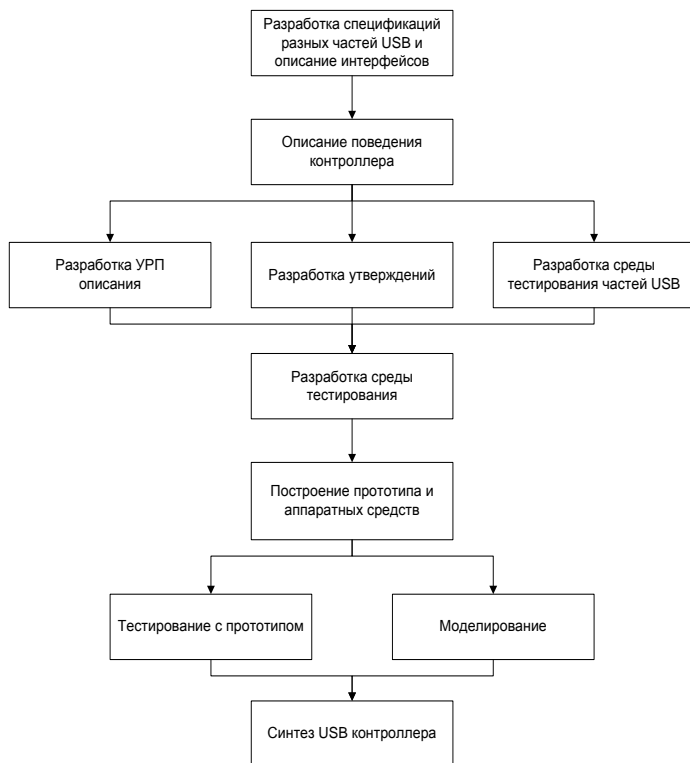


Рис. 7. Предложенная процедура построения USB контроллера

Архитектура разработанного прототипа уровня связи USB3.0 представлена на рис. 8. Как видно из рисунка, в прототипе не используется компьютер, в результате чего ограничения скорости из-за программной части не существует. Прототип состоит из приемника и передатчика данных. Тестирование может проводиться использованием только одной части логического анализатора (LeCroy, Ellysis). Таким образом, можно проверять протокол уровня связи. Прототип состоит из следующих частей:

- УРП описания уровня связи;
- физический уровень;
  - цифровая часть PCS, реализованная в ПЛИС,
  - аналоговая часть PMA, реализованная в отдельной ИС;
- блоки передачи и приема данных;
- контроллеры интерфейсов уровня связи;
- генератор случайных чисел.

Описания уровня связи и физического уровня представлены в спецификациях. Блоки тестирования внедрены для генерации и проверки данных, так как другие части USB не реализованы. Генератор случайных чисел используется для составления полезных данных проекта.

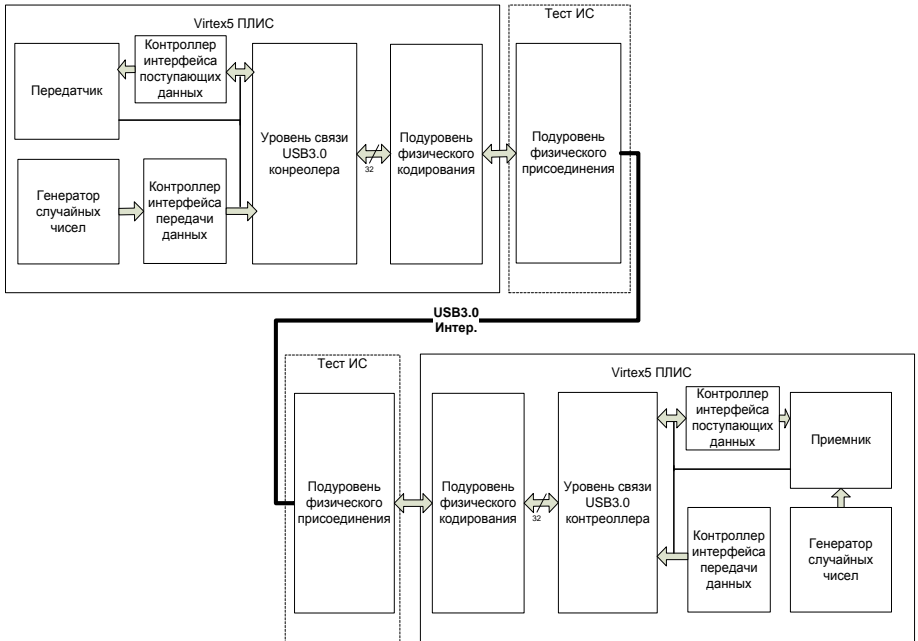


Рис. 8. Структура прототипа уровня связи USB3.0

Используются два регистра сдвига с линейной обратной связью (РСЛОС), каждый из которых генерирует 16 битов данных за один период тактового сигнала. Для построения РСЛОС используется следующий полином (рис. 9):

$$G(X) = X^{16} + X^5 + X^4 + X^3 + 1 \quad (1)$$

Аналогичный генератор используется в передатчике и в приемнике, чтобы была возможность проверки полученных пакетов.

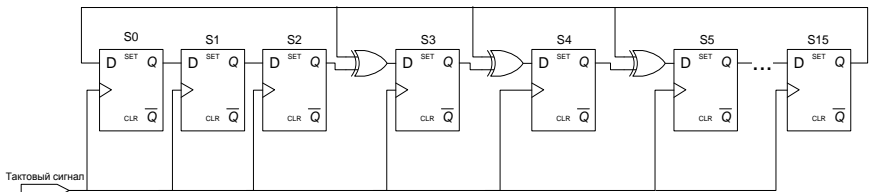


Рис. 9. Структура РСЛОС для генераций 16 битов данных

Основные конечные автоматы приемника и передатчика представлены на рис.10 а и б соответственно. Передатчик отвечает за генерацию и поставление данных к интерфейсу передачи данных уровня связи USB3.0 контроллера.

Приемник наблюдает за соответствующим интерфейсом уровня связи и проверяет полученные полезные данные.

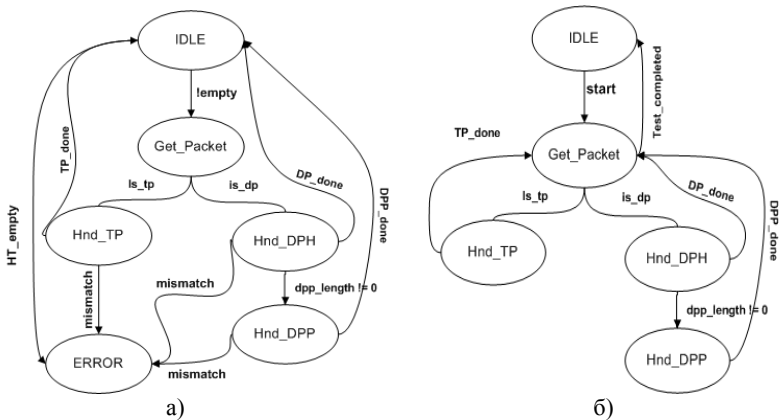


Рис. 10. Основные конечные автоматы приемника (а) и передатчика (б).

Как известно, каждый сигнал, проходящий через разные области тактового сигнала, должен быть синхронизирован, и полная проверка логики синхронизаций - большая проблема для современных ИС. Утверждения можно использовать и для тестирования цифровых ИС с несколькими тактовыми сигналами с RV системах.

**В третьей главе** на основе описанных в предыдущей главе методов разработана программная аппаратная среда TAID Instrumentor для построения прототипа и Debugger для контролирования аппаратной части во время работы прототипа. Процедура построения прототипа заключается в следующем (рис. 11):

1. В соответствии с предложенным методом синтезируемые утверждения должны быть внедрены и протестированы во время разработки контроллера. В случае тестирования синхронизирующих блоков утверждения можно внедрить автоматически, предоставив список сигналов, проходящих в области тактового сигнала, и метод синхронизаций каждого перехода. Этот список можно сгенерировать с помощью программы Spyglass CDC, которая применяется для структурного тестирования ИС с несколькими тактовыми сигналами. На основе этого списка TAID может внедрить утверждения.
2. После внедрения блоков утверждений TAID анализирует описание уровня регистровых передач и внедряет блоки контроля интерфейса среды тестирования и компьютера.
3. Как было сказано выше, УПП USB контроллера должен быть приспособлен к реализации с помощью ПЛИС технологии. Проводятся следующие модификации:
  - изменяются методы генераций тактового сигнала;
  - запоминающие устройства заменяются эквивалентными;

- триггеры, контролируемые уровнем и схемой асинхронной задержки, заменяются или удаляются;
- аналоговые части удаляются, и в случае использования аналоговых ИС внедряются контроллеры соответствующих интерфейсов.

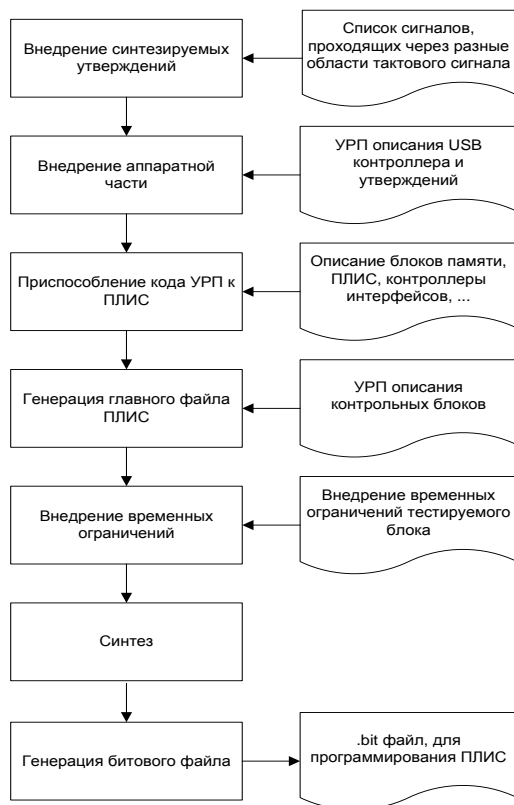


Рис.11. Процедура построения прототипа с TAID Instrumentor

Для большей части этих изменений TAID Instrumentor использует программную среду ProtoCompiler. Так как эти изменения невозможно полностью выполнить программными методами, УРП должен быть приспособлен к ПЛИС.

4. Генерируются главные описания ПЛИС, которые содержат USB контроллер и другие части, нужные для связи с компьютером и контроля среды тестирования. Генерируются также все контроллеры интерфейса прототипа и компьютера. Если система не содержит компьютер, генерируются контроллеры для связи с JTAG интерфейсом (VIO или ICON).
5. Внедряются временные ограничения USB контроллера и среды тестирования. На этом этапе вводы/выводы контроллера присоединяются к соответствующим вводам/выводам ПЛИС. Временные ограничения и конфигурация

вводов/выводов предоставляются в файле формата sdc, если для синтеза используются программные инструменты Synopsys, или xdc в случае использования программ Xilinx.

6. Проводится синтез с помощью Synplify Premier или ISE (Vivado для седьмого поколения ПЛИС).

Генерируется битовое описание прототипа, которое используется для программирования ПЛИС. Сгенерированный битовый файл содержит описания USB контроллера и среды тестирования. Чипы ПЛИС на платформе построения прототипа программируются этим битовым файлом.

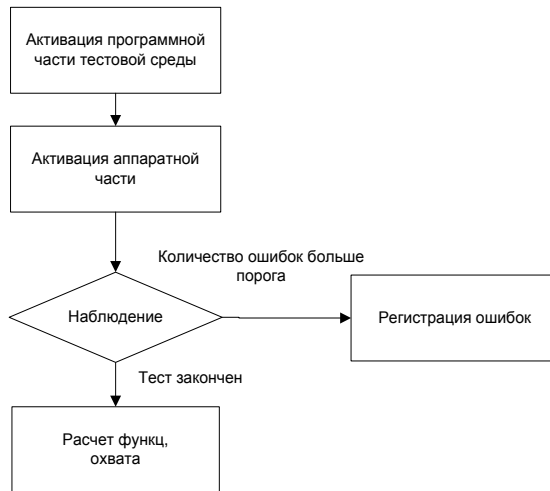


Рис. 12. Процедура TAID Debugger во время работы прототипа

Во время работы прототипа для контроля за статусом утверждений и для вывода данных применяется программа TAID Debugger. При этом используется следующая процедура (рис.12):

1. Активируется программа с помощью команды “insmod TAID\_debugger.ko”
2. Программа активирует аппаратную часть, записав соответствующие значения в регистрах. Также программируется допустимое количество ошибок.
3. Начинаются наблюдения за статусами утверждений.
4. В случае превышения допустимого количества ошибок наблюдения заканчиваются и выводятся результаты тестирования.
5. Если тестирование заканчивается и количество ошибок ниже допустимого уровня, проводится расчет функционального охвата (ФО).

Графическая часть программы TAID Instrumener предоставляет пользователю удобный интерфейс для выполнения шагов построения прототипа, представленных выше. После запуска программы появляется окно контроля проектов, с помощью которого можно создать новый проект или открыть один из текущих проектов (рис. 13).

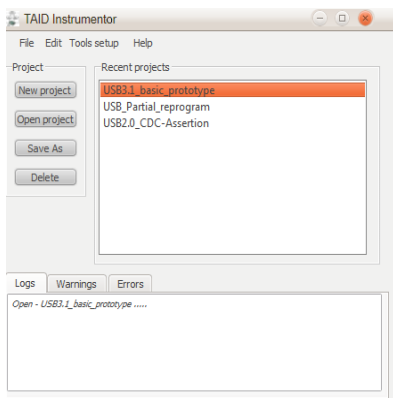


Рис. 13. Окно контроля проектов TAID Instrumentor

После выбора проекта открывается основное рабочее пространство. Программа “TAID flow” включает все представленные шаги построения прототипа и дополнительный шаг “Setup design” для выбора платформы и добавления УПП контроллера (рис. 14).

Чтобы представить применимость данного метода, надо изучить влияние предложенной среды тестирования на использование дополнительных ресурсов и скорость прототипа. Количество ресурсов, использованных USB3.0 прототипом без утверждений, и ресурсов построенного прототипа представлено в табл. 1. В случае использования предложенного метода количество Slice LUT увеличилось на 14%, а количество регистров - на 16%.

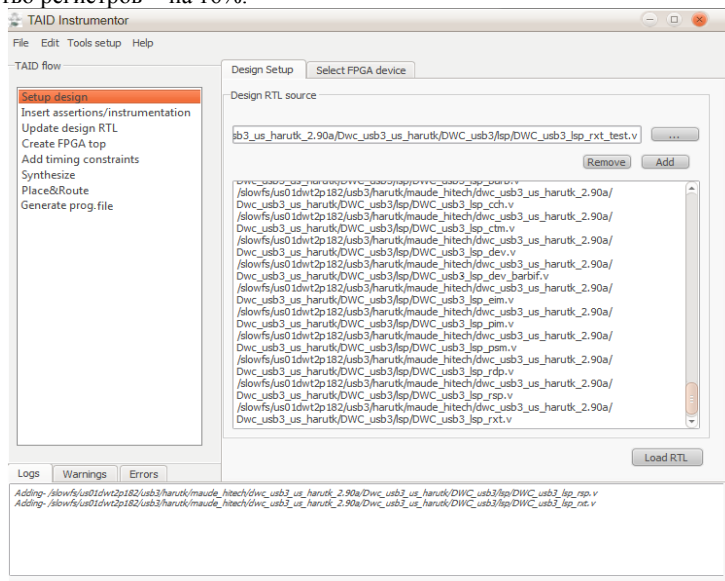


Рис. 14. Основное рабочее окно TAID Instrumentor



В таблице показано, какая часть ресурсов ПЛИС использована. Как видно из таблицы, количество ресурсов, использованных для построения прототипа по предложенному методу, меньше на 50% от общего числа ресурсов ПЛИС.

Таблица 1

*Использованные ресурсы ПЛИС*

ПЛИС ресурсы	USB3.0 прототипа	Ресурсы предложенного метода	Часть исп. ресурсовXC7VX690T (%)
Slice LUT	145174	165498	38,2
Slice register	65628	76334	8,8
F7/F8 mux	2331	2643	1,2
RAMB36	97	103	7,1
RAMB18E1	6	6	0,2
BUFGCTRL	19	21	59,38
MMCME2_ADV	4	4	20
BUFR	2	2	2,5

Применимость метода ограничивается влиянием на скорость прототипа (табл. 2). Как видно из таблицы, частота тактовых сигналов bus\_clk и pipe\_pclk чуть меньше требуемой, но это допустимо для прототипов, построенных на основе ПЛИС. Во время работы прототипа не было отмечено уменьшения скорости передачи данных.

Таблица 2

*Влияние предложенного метода на скорость прототипа*

Тактовый сигнал	Требуемая частота (МГц)	Частота без среды тестирования (МГц)	Частота внедренной аппаратной части (МГц)
bus_clk	125	126,8	117,9
pcie_pipe_clk	250	347,5	347,5
pipe3_pclk_in	125,0	169,5	160,3
pipe_pclk	125	144,3	123
rclk	100	234	216
utmi_clk	60	61,4	60,9
suspend_clk_32k	1	814	814

## ОСНОВНЫЕ ВЫВОДЫ ПО ДИССЕРТАЦИОННОЙ РАБОТЕ

1. Разработана новая процедура построения цифровой части контроллера USB с использованием синтезируемых утверждений, которая увеличивает видимость внутренних сигналов в РВ системах. Внедренные утверждения значительно упрощают процесс нахождения причин ошибок. По сравнению с другими методами, для эффективного использования которых необходимы знания о реализации проекта, предложенный метод не требует этих знаний и достаточно знать только спецификации USB интерфейса [1, 3].
2. Предложено использовать утверждения, внедренные в процессе проектирования, для описания качества тестирования прототипа с целью расчета функционального охвата [1, 3].

3. Разработан метод построения РВ системы тестирования уровня связи USB3.0 интерфейса для решения основных задач функционального тестирования. Предложенная система содержит только аппаратную часть, и поэтому ограничений скорости передачи данных USB интерфейса, связанных с программной частью, не существует. Это позволяет при тестировании получать скорость передачи данных, максимально приближенную к представленной в спецификации [2].
4. Разработан основанный на РВ системе прототип для исчерпывающего тестирования цифровых проектов с несколькими областями тактовой частоты. Синтезируемые утверждения внедрены в блоки синхронизаций сигналов, проходящих через разные области тактового сигнала для нахождения ошибок. В отличие от программных решений, использующих метод структурного анализа, предложенный метод функционального тестирования не имеет недостатков современных методов [4, 5].
5. Разработан программно-инструментальная среда TAID Instrumentor для построения прототипа, аппаратной части тестирования и внедрения синтезируемых утверждений. TAID Debugger (приложение программной части USB интерфейса) используется во время тестирования прототипа для наблюдения за сигналами, активации аппаратной части среды тестирования и считывания результатов наблюдения утверждений. Прототип USB3.0, который был создан с помощью программы TAID Instrumentor для расчета функционального охвата, когда все утверждения внедрены, использует дополнительных 14% элементов для построения таблиц поиска и 16% регистров, оказывая незначительное влияние на скорость прототипа. Чтобы уменьшить количество дополнительных ресурсов и увеличить скорость, используется возможность частичного перепрограммирования ПЛИС.

**Основные результаты диссертации** опубликованы в следующих работах:

1. **Krrikyan H.** USB IP development flow based on Synthesizable Assertions // 10th International Conference on Semiconductor Micro- and Nanoelectronics, September 11-13, 2015.- Yerevan, Armenia, 2015.- P.147-152.
2. **Krrikyan H.** Data Generator and Synthesizable Monitor for USB3.0 Link Layer Testing // 10th International Conference on Semiconductor Micro- and Nanoelectronics, September 11-13, 2015.- Yerevan, Armenia, 2015.- P.141-146.
3. **Krrikyan H., Hovhannisyan T., Manukyan S.** Prototyping system for USB3.0 link layer using synthesizable assertions and partial reconfiguration // IEEE Computer Science and Information Technologies (CSIT).- Yerevan, September, 2015.- P. 238-241.
4. **Melikyan V., Krrikyan H., Hovhannisyan T., Babayan D.** Synthesizable Assertions for Clock Domain Crossing Verification for the USB Prototype // Proceedings of the Republic of Armenia National Academy of Sciences and National Polytechnic University of Armenia. Series of Technical Sciences.- 2016.-V.69, N 2.- P. 138-150.
5. **Krrikyan H.** Functional verification method of synchronization schemes within USB2.0 prototype // Proceeding of Engineering Academy of Armenia (PEAA).- 2016.-V.13, N 1.- P. 143-146.

## ԱՄՓՈՓԱԳԻ

Թվային ինտեգրալ սխեմաների (ԻՍ) զարգացումը հանգեցրել է ավելի մեծ, բազմաթիվ մտավոր սեփականություն հանդիսացող թվային սխեմաների օգտագործմամբ, բազմաթիվ տարբեր սինքրոագրանշանների տիրույթներով և պրոցեսորներով նախագծերի տարածմանը: Մեծ տարածում են գտել նաև կիսահաղորդչային բազմապրոցեսոր համակարգերի (ԿԲՊՀ) տիպի նախագծերը: Այս ամենը հանգեցրել է թեստավորման տևողության զգալի աճի (2014 թվականին նախագծերի քանակը, որոնց թեստավորումը կազմում է նախագծման տևողության 80%-ից ավելին, զգալիորեն աճել է): Թեստավորման տևողության և թեստավորող ճարտարագետների թվի աճի կարգավարումը շատ կարևոր խնդիր է ժամանակակից թվային նախագծերում: Բացի դրանից, ժամանակակից թվային նախագծերում ծրագրային մասի նախագծումը գրեթե նույնքան ժամանակատար է, ինչ ապարատային մասինը: Սովորական հաջորդական նախագծման դեպքում, երբ ծրագրային մասի նախագծումը սկսվում է առաջին ԻՍ-ի արտադրությունից հետո, ծրագրային մասի նախագծման ժամանակատարության պատճառով հնարավոր է, որ նախատեսված ժամանակահատվածում նախագծի վաճառքում հայտնվելու հարցում խնդիրներ առաջանան: Անհրաժեշտ է մշակել նոր միջոցներ թեստավորման որակի և արդյունավետության բարձրացման նպատակով:

Վերակառուցավորվող հաշվողական (ՎՀ) համակարգերի կիրառմամբ կառուցված նախատիպերի թեստավորման գործառնությունների աճը կարող է զգալիորեն նվազեցնել թեստավորման տևողությունը և բարձրացնել դրա որակը: Ծրագրային մասի նախագծումը կարելի է սկսել առաջին ֆունկցիոնալ նախատիպի առկայության դեպքում: Ծրագրավորվող տրամաբանական ինտեգրալ սխեմաների (ԾՏԻՍ) զարգացումը տվել է ավելի մեծ ԻՍ-ների նախատիպերի կառուցման հնարավորություն (մեկ ԾՏԻՍ-ում հնարավոր է կառուցել 4 միլիոն տարբեր պարունակող ԻՍ-ի նախատիպ): Բացի դրանից, ժամանակակից ԾՏԻՍ-ների մասնակի վերակառուցավորման հատկությունը կարող է նոր տիպի թեստավորման համակարգերի կառուցման հնարավորություն տալ:

ԾՏԻՍ նախատիպերով թեստավորման հիմնական խնդիրներից են սխալների հայտնաբերումը և տեղայնացումը: Այս գործընթացը, նախատիպի ներքին ազդանշանների տեսանելիության բացակայության պատճառով, իր բարդությամբ համադրելի է կիսահաղորդչում սխալների հայտնաբերմանը: Արդի ՎՀ համակարգերի կառուցման մեթոդները նախատիպի ներքին ազդանշանների տեսանելիության բարձրացման տեսանկյունից ամբողջությամբ չեն օգտագործում ԾՏԻՍ-ի զարգացման արդյունքում ստացված հնարավորությունները: Բացի դրանից, ԾՏԻՍ նախատիպի թեստավորման որակի գնահատման համար շատ կարևոր ցուցանիշի՝ ֆունկցիոնալ ծածկույթի, հաշվարկի միջոցը, որն ընդունվում է բոլորի կողմից, բացակայում է:

Արդի մեթոդները հնարավորություն են տալիս ազդանշանները դիտարկել միայն շատ փոքր տևողությամբ: Այդ նպատակով կարելի է կառուցել ՎՀ համակարգեր, որոնք սինթեզվող հաստատումների միջոցով կկատարեն նախատիպի ներքին ազդանշանների դիտարկումը համակարգի աշխատանքի ընթացքում՝ առանց ընդհատման:

Բացի վերը նշվածից, ՎՀ համակարգերը կարելի է օգտագործել նաև մի քանի սինքրոագրանշանով նախագծերի թեստավորման որակի բարձրացման նպատակով: Պարզապես դիտարկելով նախատիպի աշխատանքը՝ կարելի է ստուգել սինքրոագրանշանի տարբեր տիրույթներով անցնող ազդանշանների սինքրոագման տրամաբանական հանգույցները: Սակայն, սխալ վարքի դեպքում գրեթե հնարավոր չէ պարզել պատճառը:

Ատենախոսությունը նվիրված է վերը նկարագրված խնդիրների լուծման համար ՎՀ համակարգերի կառուցման միջոցների մշակմանը:

Մշակվել է համապիտանի հաջորդական դողի (<<Դ) դեկավարման թվային հանգույցի նախագծման նոր երթուղի՝ սինթեզվող հաստատումների կիրառմամբ, որը ՎՀ համակարգի ներքին ազդանշանների տեսանելիության բարձրացման հնարավորություն է տալիս: Ներդրված հաստատումները զգալիորեն պարզեցնում են սխալների պատճառների հայտնաբերման գործընթացը: Ի տարբերություն այլ մեթոդների, որոնց արդյունավետ կիրառման համար անհրաժեշտ են գիտելիքներ նախագծի իրականացման վերաբերյալ, առաջարկված մեթոդը չի պահանջում դրանք, և գիտելիքները միայն <<Դ ինտերֆեյսի տեխնիկական արձանագրության վերաբերյալ բավարար են [1,3]:

Առաջարկվել է նախագծման ընթացքում ներդրված հաստատումների կիրառումը՝ նախատիպի թեստավորման որակի գնահատման ֆունկցիոնալ ծածկույթի հաշվարկի նպատակով [1,3]:

Մշակվել է <<Դ3.0 ինտերֆեյսի կապի մակարդակի թեստավորման ՎՀ համակարգի կառուցման միջոց՝ ֆունկցիոնալ թեստավորման հիմնական խնդիրների լուծման նպատակով: Առաջարկված համակարգը պարունակում է միայն ապարատային մաս, այդ իսկ պատճառով՝ <<Դ ինտերֆեյսի տվյալների փոխանցման արագության ծրագրային սահմանափակումները բացակայում են: Թեստավորման ընթացքում դա թույլ է տալիս ստանալ տվյալների փոխանցման այնպիսի արագություն, որն առավելագույնս մոտ է տեխնիկական արձանագրությունում նշվածին [2]:

Մշակվել է մի քանի սինքրոազդանշաններով թվային նախագծերի ամբողջական թեստավորման նախատիպ՝ հիմնված ՎՀ համակարգի վրա: Սինթեզվող հաստատումները ներդրվում են սինքրոազդանշանի տարբեր տիրույթների միջև անցում կատարող ազդանշանների սինքրոնացման տրամաբանական հանգույցներում՝ սխալների հայտնաբերման նպատակով: Ի տարբերություն ծրագրային միջոցների, որոնք կատարում են նախագծի կառուցվածքային վերլուծություն, առաջարկվող մեթոդը կատարում է ֆունկցիոնալ թեստավորում նախատիպում, հետևաբար՝ արդի մեթոդների թերություններն այս մեթոդում բացակայում են [4,5]:

Նախատիպի կառուցման, թեստավորման ապարատային մասի և սինթեզվող հաստատումների ներդրման նպատակով նախագծվել է TAID Instrumentor ծրագրային գործիքային միջավայրը: TAID Debugger <<Դ ծրագրային ապահովման հավելվածն օգտագործվում է նախատիպի թեստավորման ընթացքում ազդանշանների դիտարկման, արդյունքների դուրս բերման և հաստատումների դեկավարման նպատակով: TAID Instrumentor ծրագրային միջոցով կառուցված <<Դ3.0 նախատիպը վատագույն դեպքում, երբ ներդրված են ֆունկցիոնալ ծածկույթի հաստատումները, սպառում է 14%-ով ավելի իսկության արդյունավետ կառուցման նպատակով օգտագործվող տարրեր և 16%-ով ավելի տրիգերներ: Բացի այդ, նախատիպի արագագործությունն աննշան նվազում է: Սակայն, այդ արդյունքն ընդունելի է ՄՏԻՍ-ով կառուցված նախատիպի դեպքում: Արագագործության բարձրացման և հավելյալ ռեսուրսների նվազեցման նպատակով օգտագործվել է ՄՏԻՍ-ի մասնակի ծրագրավորման հատկությունը:

## **HARUTYUN RUBEN KRRIKYAN**

### **DEVELOPMENT AND INVESTIGATION OF RECONFIGURABLE COMPUING SYSTEM ARCHITECTURES**

#### **SUMMARY**

The recent development of digital integrated circuits (IC) has resulted in a wider spread of big designs with multiple IP blocks, clock domains, and processors. The system on a chip (SoC) - type designs have become more common. As a result, the duration of testing has increased significantly. It is worth mentioning that in 2014 the number of designs in which at least 80% of the duration of development was spent on testing, has increased. The control of the testing time and the number of engineers involved in testing is an important issue in modern digital designs. Besides, the software development in modern digital designs consumes almost as much time as the development of hardware. In case if the traditional sequential development begins after the production of the first IC, problems may arise concerning the appearance of the design in the market in the anticipated period of time because of the long duration of the development. It is necessary to develop new methods to increase the testing quality and the efficiency.

The increase in the testing functionality of prototypes based on reconfigurable computing systems, may significantly enhance the quality, and decrease the duration of testing. In case of existing of the first functional prototype, the development of the software can be started. The recent development of field programmable gate array (FPGA) has made it possible to create bigger IC prototypes with about 4 million application specific integrated circuit (ASIC) equivalent gates. Besides, the partial/dynamic re-programmability of FPGA provides an opportunity to build new types of testing environments.

One of the main issues of the FPGA prototype testing is the detection and localization of errors. The difficulty of this process is comparable with the same process in the silicon because of the small visibility of the internal signals of FPGA. Modern reconfigurable computing (RC) systems do not use the features of FPGA completely from the point of view of increasing the visibility. A commonly accepted calculation method for the quality testing i.e. the functional coverage for the prototype does not exist.

Modern methods allow monitoring the signals only for a short period of time. For that purpose, RC systems can be developed that can observe the internal signals of the prototype by the synthesized assertions during the operation of the system without interruption.

Apart from this, the RC system can be used for complete testing of the clock domain crossing (CDC) signal synchronization logic. By simply observing the operation of the prototype, it is possible to verify the synchronization logic for the signals passing through the clock domain, but in case of faulty behavior, the detection and localization of errors is almost impossible.

The dissertation is devoted to the development of RC systems to solve the problems mentioned above.

A new flow of development of a USB controller is proposed by using synthesizable assertions. It gives an opportunity to increase the visibility of internal signals of the RC system. Embedded assertions significantly simplify the process of debugging of errors.

The proposed method does not require knowledge on implementation of the design and knowledge of USB interface specification is enough [1, 3].

The embedded synthesizable assertions are proposed to be used for calculating the functional coverage, which is the quality estimation for the testing with the prototype [1, 3].

An architecture of testing environment for USB 3.0 link layer is proposed to solve the main problems of functional testing. The proposed system includes only hardware parts, as a result the speed limitations imposed by software do not exist. This provides an opportunity to get speed of data transfer maximally close to value provided in the specification [2].

A method for testing the CDC synchronization logic by using assertions is developed based on RC system. The synthesizable assertions are embedded into synchronization logic of CDC signals and are used for debugging of errors. In comparison with the other methods which use structural analysis of the design, the proposed method performs functional testing within the prototype, eliminating the drawbacks of the modern methods [4, 5].

TAID Instrumentor software and hardware environment was developed to build the prototype, to insert testing hardware and synthesizable assertions. TAID Debugger, which is an addition to USB driver, is used for monitoring signals during the operation of the prototype, for retrieving results, and for controlling the assertions. USB3.0 prototype, which is built with TAID Instrumentor, consumes additional 14% lookup tables and 16% registers in the worst case, when all assertions for functional coverage are inserted. The speed of the prototype decreases insignificantly which is acceptable for prototypes built with FPGA. To increase the speed and to decrease the amount of additional resources partial programmability of FPGA has been used.

