

ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ԿՐԹՈՒԹՅԱՆ, ԳԻՏՈՒԹՅԱՆ,
ՄՇԱԿՈՒՅԹԻ ԵՎ ՄՊՈՐՏԻ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

Հարությունյան Ստեփան Արթուրի

ԲԱՐՁՐ ԱՐԱԳԱԳՈՐԾՈՒԹՅԱՄԲ ԹՎԱՅԻՆ ԻՆՏԵԳՐԱԼ ՄԽԵՄԱՆԵՐԻ
ՆԱԽԱԳԾԵՐԻ ՍՏՈՒԳՄԱՆ ՄԻՋՈՑՆԵՐԻ ՄՇԱԿՈՒՄԸ

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

ՍԵՂՍԱԳԻՐ

Երևան 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ, НАУКИ, КУЛЬТУРЫ И СПОРТА
РЕСПУБЛИКИ АРМЕНИЯ

НАЦИОНАЛЬНЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ АРМЕНИИ

Արտյունյան Տեփան Արտուրովիչ

РАЗРАБОТКА СРЕДСТВ ВЕРИФИКАЦИИ ПРОЕКТОВ
БЫСТРОДЕЙСТВУЮЩИХ ЦИФРОВЫХ ИНТЕГРАЛЬНЫХ СХЕМ

АВТОРЕФЕРАТ

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

Երևան 2022

Ատենախոսության թեման հաստատվել է Հայաստանի ազգային պոլիտեխնիկական համալսարանում (ՀԱՊՀ):

Գիտական ղեկավար՝ տ.գ.դ. Վազգեն Շավարշի Մելիքյան

Պաշտոնական ընդդիմախոսներ՝ տ.գ.դ. Սուրիկ Խաչիկի Խուդավերդյան
տ.գ.թ. Նարեկ Էդուարդի Մամիկոնյան

Առաջատար կազմակերպություն՝ Երևանի Պետական Համալսարան

Ատենախոսության պաշտպանությունը կայանալու է 2022թ. հուլիսի 8-ին, ժամը 16⁰⁰-ին, ՀԱՊՀ-ում գործող «Ռադիոտեխնիկայի և էլեկտրոնիկայի» 046 Մասնագիտական խորհրդի նիստում (հասցեն՝ 0009, Երևան, Տերյան փ., 105, 17 մասնաշենք) :

Ատենախոսությանը կարելի է ծանոթանալ ՀԱՊՀ– ի գրադարանում:

Սեղմագիրն առաքված է 2022թ. մայիսի 25-ին:

046 Մասնագիտական խորհրդի
գիտական քարտուղար. տ.գ.թ.



Բենիամին Ֆելիքսի Բադալյան

Тема диссертации утверждена в Национальном политехническом университете Армении (НПУА)

Научный руководитель: д.т.н. Вазген Шаваршевич Меликян

Официальные оппоненты: д.т.н. Сурик Хачикович Худавердян
к.т.н. Нарек Эдуардович Мамиконян

Ведущая организация: Ереванский Государственный Университет

Защита диссертации состоится 8-го июля 2021 г. 16⁰⁰ ч. на заседании Специализированного совета 046 — "Радиотехники и электроники", действующего при НПУА, по адресу: 0009, г. Ереван, ул. Теряна, 105, корпус 17.

С диссертацией можно ознакомиться в библиотеке НПУА.

Автореферат разослан 25-ого мая 2022 г.

Ученый секретарь
Специализированного совета 046, к.т.н.



Бенямин Феликсович Бадалян

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Параллельно с развитием современных интегральных схем (ИС) увеличивается и число используемых в схемах логических элементов, достигая нескольких миллиардов. Каждый добавленный в проектируемую схему компонент требует дополнительного времени для верификации. В последние годы в связи с темпами увеличения размерности схем, часто 70...80% продолжительности проекта занимает его верификация. Для обеспечения высокого качества покрытия необходимо проверить все возможные входные комбинации, в противном случае получатся пробелы в покрытии, вызывающие проблемы качества и надежности продукции. Исходя из вышеизложенного, затрачивается много времени для организации процесса верификации посредством покрытий. Достижение максимального значения покрытий - чрезвычайно долгий процесс. Необходимо создать особые последовательности проверки и моделировать их. Некоторые виды покрытий требуют моделирование каждой строки проекта для получения максимального значения. На рынке вводятся стандарты качества, согласно которым минимальное значение качества покрытий должно составить 95...98%, причем обеспечить это значение необходимо очень быстро, чтобы остаться конкурентоспособным. В условиях таких вызовов существующие методы не удовлетворяют поставленным требованиям, в связи с чем необходимо разработать новые быстродействующие средства верификации.

Таким образом, разработка средств верификации проектов быстродействующих цифровых интегральных схем является весьма актуальным. Это обусловлено прежде всего тем, что известные из литературы методологии, методы, подходы, решения и архитектуры не удовлетворяют требованиям, возникаемым с ростом сложности современных проектов, что, в свою очередь, вызывает необходимость разработки новых подходов для решения вышеуказанных задач.

Диссертация посвящена разработке средств верификации быстродействующих цифровых ИС, ускоряющих процесс проверки компонентов ИС и тем самым обеспечивающих высокое качество продукта.

Предмет исследования. Методы увеличения быстродействия средств верификации проектов цифровых ИС с целью сокращения времени, потраченной на проверку.

Цель работы. Разработка средств верификации проектов быстродействующих цифровых ИС.

Методы исследования. При проведении исследования были использованы методологии и подходы к созданию среды верификации, известные методы квалификации, улучшения и создания средств верификации, теории эффективного использования процесса проверки и его компонентов.

Научная новизна:

- Предложены способы разработки средств верификации проектов быстродействующих ИС, которые значительно улучшают их быстродействие, сокращают время выявления погрешности и продолжительность проектирования, повышают качество проверки.

- Разработан метод динамического управления проверочными точками средств верификации проектов быстродействующих цифровых ИС, который благодаря уменьшению числа повторяющихся измерений проверочных точек и аргументов, обеспечивает рост быстродействия последовательностей проверок на 3...5% за счет усложнения архитектуры среды.
- Создан метод формирования среды верификации проектов быстродействующих цифровых ИС, который благодаря автоматической генерации компонентов обеспечивает уменьшение длительности процесса верификации в среднем на 42,7% за счет снижения точности работы функциональных блоков всего лишь на 10...15%.
- Предложен метод улучшения характеристик проверочной интеллектуальной собственности (ПИС) средств верификации проектов быстродействующих цифровых ИС, который благодаря измененной архитектуре и внедрению последовательностей «начало-пауза-продолжение» обеспечивает повышение скорости процесса верификации почти на 50% и увеличение быстродействие проверочных последовательностей на 5...8%, за счет добавления компонентов в среду.

Практическая ценность работы. Разработан программный инструмент “TE Generator” для создания среды верификации проектов быстродействующих цифровых ИС, который имеет удобный для пользователя графический интерфейс и дает возможность создавать проверочную среду с использованием методологий “Универсальная методология верификации” или “Руководство по методике верификации”, а также генерировать среду улучшенной ПИС. Аprobация программного обеспечения (ПО) в ряде настоящих проектов показала его высокую эффективность по сравнению с ручной разработкой среды верификации: за счет уменьшения точности проверочной среды всего на 10...15% обеспечивается сокращение срока проектирования в среднем на 40...70%.

Достоверность научных положений Научные результаты, приведенные в диссертации, подтверждены математическими обоснованиями в сочетании с результатами практического моделирования.

Внедрение. ПО “TE Generator” внедрено в ЗАО “Синописис Армения”. Оно используется для создания среды верификации, выполнения изменений различных существующих решений, создания улучшенной ПИС, а также для изучения полученных результатов. Предложенные быстродействующие методы внедрены в среду верификации, созданной ПО “TE Generator”, показано их эффективность в течении производственного процесса..

Основные положения, выносимые на защиту:

- метод динамического управления проверочными точками средств верификации проектов быстродействующих цифровых ИС
- метод формирования среды верификации проектов быстродействующих цифровых ИС
- метод улучшения характеристик ПИС средств верификации проектов быстродействующих цифровых ИС.

Аprobация работы. Основные научные и практические результаты диссертации докладывались на:

- 39-й Международной конференции "ELECTRONICS AND NANOTECHNOLOGY" (Киев, Украина, 2019 г.);
- 40-й Международной конференции "ELECTRONICS AND NANOTECHNOLOGY" (Киев, Украина, 2020 г.);
- 19-ом Международном симпозиуме "IEEE EWDTS: East-West Design & Test " (Батуми, Грузия, 2021 г.);
- научных семинарах кафедры "Микроэлектронные схемы и системы" НПУА (Ереван, Армения, 2019-2022 гг.);
- научных семинарах ЗАО "Синописис Армения" (Ереван, Армения, 2019 - 2022 гг.).

Публикации. Основные положения диссертации опубликованы в девяти научных трудах, список которых приведен в конце автореферата.

Структура и объем работы. Диссертация состоит из введения, трех глав, основных выводов, списка литературы, включающего 102 наименования, а также 5 приложений. В первом приложении вложен акт внедрения результатов диссертации, во втором – отрывок проверочной среды интерфейса “Универсальный асинхронный приемопередатчик”, в третьем – отрывок описания графического интерфейса программного инструмента “TE Generator”, в четвертом – список рисунков и таблиц, в пятом – список использованных сокращений. Основной текст диссертации составляет 104 страниц, а вместе с приложениями – 160 страниц, включая 66 рисунка и 9 таблиц. Диссертация написана на армянском языке.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертационной работы, сформулирована цель исследования, представлены научная новизна, практическая значимость и основные научные положения, выносимые на защиту.

В первой главе проведен обзор литературы и приведены требования к среде верификации цифровых ИС. Рассмотрены наиболее распространенные методы верификации ИС и дана оценка их эффективности. Показано, что существующие методы не поспевают за темпами роста размеров современных ИС.

Современные цифровые ИС содержат огромное количество транзистров и с развитием технологий эта тенденция сохраняется. Одновременно с увеличением количества функциональных блоков вырастает время, необходимое для верификации ИС. В среднем на верификацию ИС затрачивается больше половины времени всего проекта по разработки ИС, а иногда - почти 80% (рис 1).

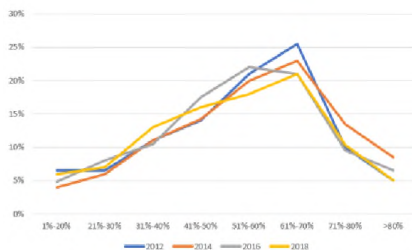


Рис. 1. Время потраченное на верификацию проекта

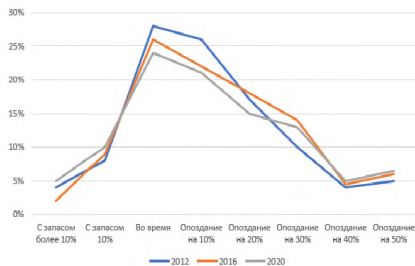


Рис. 2. Время завершения проекта

Использование такого большого количества времени не всегда достаточно для полной уверенности, что все компоненты ИС работают правильно, и приходится затрачивать дополнительное время для завершения верификации. Исследования показывают, что только 20% проектов выполняются в назначенный срок, а в остальных случаях тратится до 50% запланированного времени (рис. 2).

Проект считается завершенным, когда так называемые функциональные точки выполнены. Эти точки выбираются во время планирования и представляют собой интерпретацию документации продукции. С повышением числа функциональных блоков число обязательных точек тоже растет. Традиционный подход для получения полного покрытия функциональных точек требует большого количества последовательностей. Как правило, для одной точки пишется одна последовательность. Этот метод может исчерпать себя, когда число функциональных точек достигает нескольких сотен или тысяч.

Из приведенных примеров и аргументов становится ясно что проблема верификации больших ИС требует более быстроедействующих решений. Таким решением является использование контролируемой ограниченной проверки (КОП). Данный метод позволяет снизить количество необходимых последовательностей до минимума, используя случайные значения нескольких параметров. От этих параметров зависит, как будет работать последовательность. Изменяя эти параметры, получаются другие результаты.

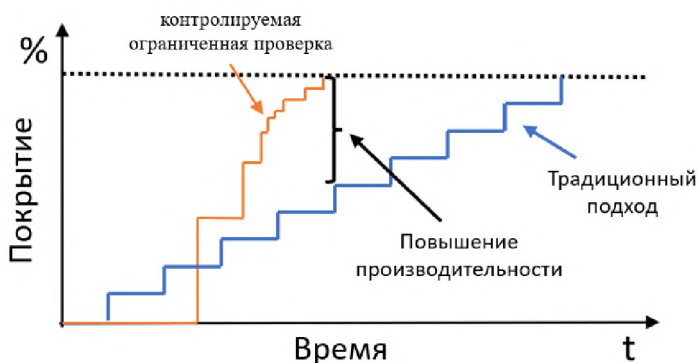


Рис. 3. Традиционная верификация и контролируемая ограниченная проверка

По сравнению с традиционным методом верификации (рис 3), в начальном интервале, метод КОП не дает ощутимых результатов. Это время тратится на разработку среды верификации. После того, как среда готова, преимущества метода КОП становятся более очевидными. Самые невероятные и сложные ситуации решаются более тонкой настройкой КОП.

Этот принцип работы среды верификации используется в разных методиках, как, например, в универсальной методике верификации (УМВ). Этот метод определяет архитектуру среды верификации и позволяет создавать компоненты, которые можно повторно использовать в других проектах. Другим преимуществом этого метода является наличие понятия «фабрика». Этот компонент позволяет менять части среды без компиляции, снижая время моделирования. Недостатком

данного метода является сложность и невозможность перехода к нему с действующей среды. Использование метода УМВ требует знаний объектно-ориентированного программирования. В частности, в ней все время используется наследование и классы. Этот метод работает с использованием так называемых фаз. Каждому этапу среды соответствует фаза, что препятствует интеграции этого метода в уже действующие среды верификации, так как невозможно синхронизировать работу компонентов.

Для уменьшения работы среды используются проверочные интеллектуальные собственности. Они заменяют собой большие части среды верификации и избавляют от необходимости вручную переходить из классов в сигналы. Недостатком метода ПИС является то, что он требует больших затрат. Для использования ПИС необходима приобретение лицензий, что повышает стоимость проекта. Кроме того, недостатками являются большое время интеграции в среду верификации и отсутствие возможности менять что-либо в самой ПИС. Как правило, это закрытые блоки и в них нельзя что-то менять. Это создает проблемы, связанные с функциональностью, немного отличающейся от стандартной.

Еще одним методом уменьшения времени верификации является строгое контролирование случайной генерации значений. Из опытов видно (рис. 4 и 5) что можно получить прирост в производительности в 36% при строгом контроле генераций параметров. Недостатком такого подхода является нестабильность результатов. Результаты меняются в зависимости от использованного симулятора. Более того, использование этого метода в средах верификации, где имеются несколько сотен параметров, не представляется возможным, так как это достаточно сложный процесс и полученный прирост производительности не стоит потраченного времени.

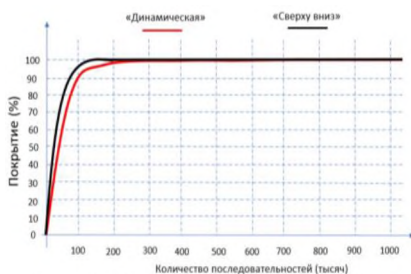


Рис. 4. Первое сравнение методов «Динамический» и «Сверху вниз»

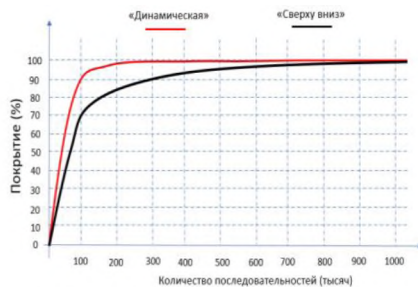


Рис. 5. Второе сравнение методов «Динамический» и «Сверху вниз»

Анализ существующих быстродействующих методов верификации ИС показал, что эти методы не имеют достаточной эффективности для проверки современных ИС. Следовательно, возникает необходимость создание новых и более быстродействующих решений верификации цифровых ИС.

Также сформулированы требования к разработке средств верификации проектов быстродействующих цифровых ИС. Показано, что процесс верификации занимает большую часть цикла создания ИС и непосредственно влияет на ее качество и цену.

Во второй главе предложены методы разработки сред верификации для быстродействующих ИС, улучшающих или исключаящих недостатки методов, представленных в первой главе.

Верификация цифровых ИС включает: создание плана проверки, формирование среды, разработку тестируемых последовательностей и получение максимального покрытия функциональных точек. Уменьшение любого из этих компонентов позволит быстрее получить результаты верификации и тем самым уменьшить стоимость разработки.

Первым из представленных методов является автоматическая генерация компонентов среды верификации. При верификации больших ИС используется КОП. На основе КОП было разработано несколько методик верификации. Эти методики имеют свои правила использования. Кроме того, они имеют определенную архитектуру. Именно эти правила и определенная архитектура используются в данном методе.

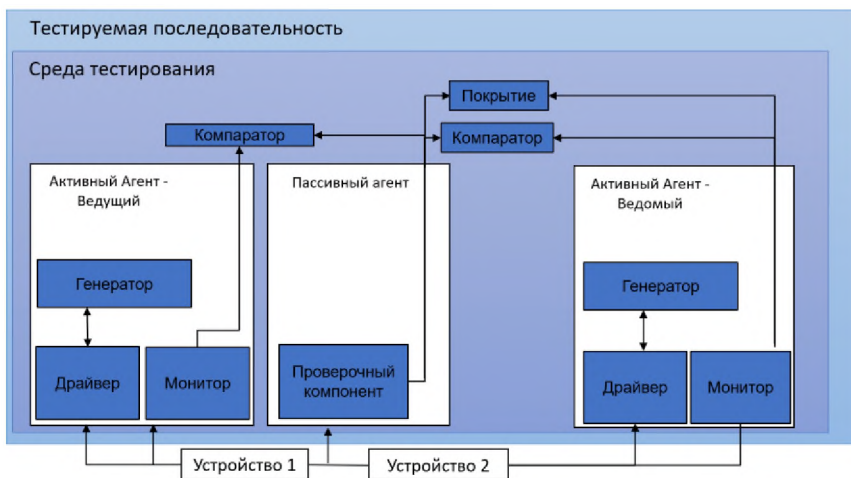


Рис. 6. Пример тестируемой среды УМВ

Метод будет рассмотрен на примере УМВ. Среда верификации УМВ (рис. 6) состоит из агентов, генераторов, драйверов, мониторов и т. д. Все эти компоненты создаются отдельно, но поскольку их будут использовать в УМВ, то они наследуются от базовых типов методики. Это позволяет автоматически генерировать их, но в полученных компонентах отсутствует функциональный код. Этот недостаток невозможно устранить так как функциональный код не подлежит генерации. Следует еще отметить, что данный метод эффективен еще тем, что снижает возможность человеческого фактора. Как было отмечено ранее, все базовые методики трудны в освоении, а с помощью этого метода возможно автоматически получать среду верификации, без необходимости составлять специфический код для методики. В отсутствующие элементы входят и проверочные блоки, в случае чего, генерированная среда теряет примерно 10...15% точности. Если разработчик добавит эти проверочные блоки вручную то потерянная точность восстановится.

Для оценки метода были разработаны среды верификации для двух равных по сложности интерфейсов. Для этих интерфейсов был разработан план верификации (табл. 1), на основе которого была проведена оценка эффективности.

Таблица 1

План верификации обоих интерфейсов

Этап верификации	Планировка верификации	Разработка среды	Разработка последовательностей	100% покрытие
Количество дней	1	20	5	10

Представленный метод разработан с целью быстрее создать среду верификации. Метод использует образцы основных компонентов методологии для автоматической генерации среды. Важной частью представленного метода является выбор методики. В данный момент метод поддерживает методики УМВ и руководство по методологии верификации (РМВ). Как видно из примера кода разработанной среды (рис. 7), вначале выбирается методика.

```
#define UVM_ENV
...

driver("ahb_driver_vip", ahb_if, "drive_item, send_address, send_data, send_stop");
generator("ahb_generator_vip", "ahb_driver_vip");
monitor("ahb_monitor_vip", ahb_if);

subenv("ahb_senv_vip", "ahb_generator_vip", "ahb_monitor_vip");

driver("ahb_driver_app", ahb_app, "drive_item, send_address, send_data, send_stop");
generator("ahb_generator_app", "ahb_driver_app");
monitor("ahb_monitor_app", ahb_app);

subenv("ahb_senv_app", "ahb_generator_app", "ahb_monitor_app");

...
```

Рис. 7. Пример генерации среды с методикой УМВ

Затем при помощи универсальных функций создается среда. В функциях есть как обязательные (например, имя компонента, физический интерфейс и т.д.) так и необязательные аргументы (дополнительные функции). В аргументах можно задавать дополнительные функции, которые будут в конечном коде. В коде нет функциональной имплементации, но имеются объявление функции и пустой блок имплементации. Это позволит снизить возможность ошибок во время кодирования. Так как методики работают по-разному, такой подход должен сделать их более доступными и легкими в освоении.

Пример приведенный на рис. 7, представляет собой готовую среду верификации (рис. 8). Для генерации с использованием данной методики был выбран интерфейс усовершенствованная высокопроизводительная шина (УВШ). Метод так-же позволяет генерировать конструкции interface на языке SystemVerilog, которые позже будет использоваться в самой среде.

Вторая среда была создана вручную. Для этой среды был выбран расширенный/расширяемый интерфейс (РРИ). Как было отмечено ранее, этот метод предназначен для снижения времени создания среды верификации. Полученные результаты (табл. 2) подтверждают эффективность методики.

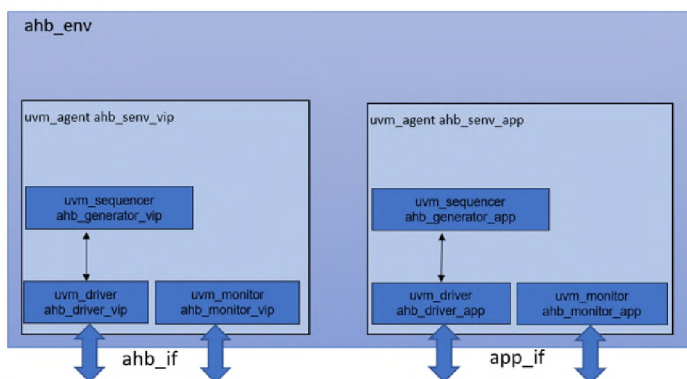


Рис. 8. Сгенерированная среда при помощи созданной методики

Как видно из результатов, этот метод влияет только на время разработки среды. С помощью этого метода время, потраченное на разработку среды снизилось почти на 75%. В целом это привело к снижению времени всей верификации интерфейса на 42%. С помощью этого метода можно также сгенерировать всю среду за несколько часов, но, как было сказано ранее, сгенерированный код не имеет функциональных блоков, а их имплементация требует времени. Также в коде отсутствуют проверочные блоки, которые снижают точность среды на 10...15%. Если после генерации среды добавить проверочные блоки вручную, то этот недостаток будет устранен.

Таблица 2

Результаты сравнения традиционного и представленного методов

Этап верификации	Запланированное время(день)	РРИ – традиционная методика (день)	УВП – представленная методика (день)
Планирование верификации	1	1	1
Разработка среды	20	19	5
Разработка последовательностей	5	5	5
100% покрытие	10	10	10
Общее	36	35	21

Вторая разработанная методика изменяет архитектуру ПИС для более быстрого получения результатов, чем обычно. Кроме измененной архитектуры, в ПИС использован специальный метод “начало-пауза-продолжение” для последовательностей.

ПИС - довольно распространенный метод в современных проектах. С помощью ПИС нет необходимости работать вручную, переделывать транзакции в физические сигналы. Как отмечалось ранее, одним из недостатков ПИС является время интеграции в среду верификации. Для работы ПИС обычно требуется специальный тип данных. Для правильного создания такого типа, их использования в последовательностях и интеграции ПИС в среду требуется довольно много времени, что задерживает изначальные результаты. Предложенная архитектура (рис. 9) добавляет в ПИС блок начальной проверки, которая ускоряет получение данных.

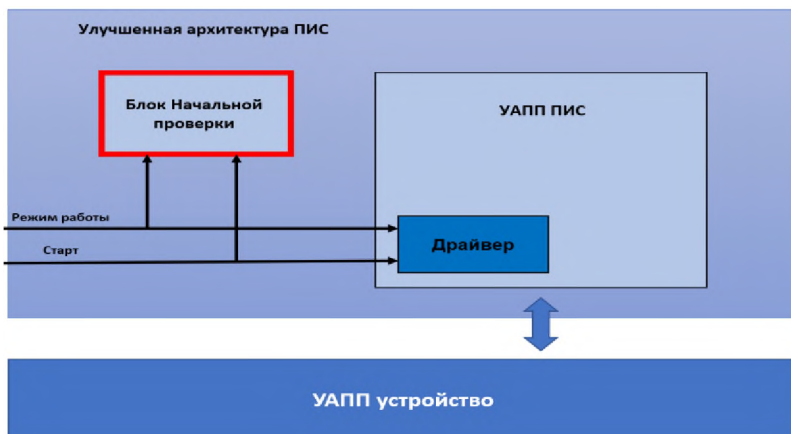


Рис. 9. Сгенерированная среда при помощи созданной методики

Работа этого компонента очень похожа на работу SCAN регистров. Компонент имеет входы «Режим работы» и «Старт». Если «Режим работы» активен (логическая 1) то в таком случае при получении пульсового сигнала на входе «Старт» драйвер автоматически получает заранее выбранные последовательности. Эти последовательности создаются или выбираются после создания ПИС. Для этого затрачивается дополнительное время, что является недостатком этого метода. Выбранные последовательности проверяют базовые функциональности интерфейса. В последовательности также интегрирована мотодика “начало-пауза-продолжение” (рис. 10).

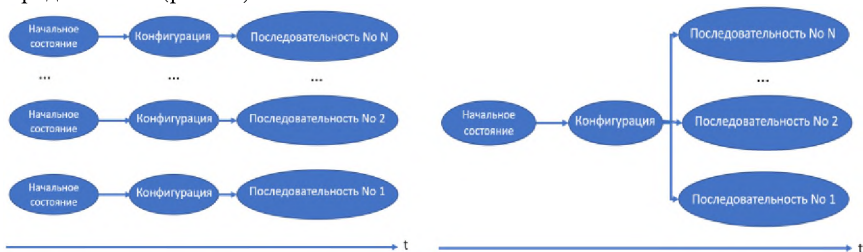


Рис. 10. Сравнение методики “начало-пауза-продолжение” с традиционным

Как видно из рис. 10, методика позволяет только один раз проходить общие части последовательности. Это имеет важное значение, поскольку любая цифровая ИС перед использованием сбрасывается, а потом конфигурируется.

Случайная генерация, использованная в среде, не является полной. Она зависит от начального значения: семени. Каждая последовательность работает с несколькими семенами для получения разных входящих данных. Именно в таких случаях методика “начало-пауза-продолжение” имеет большую эффективность.



Рис. 11. Пример работы последовательности

Для оценки метода была разработана среда с использованием улучшенной ПИС и без ПИС. Были использованы 9 последовательностей для оценки времени (рис. 11). Из данных табл. 3 видно, что при использовании улучшенной ПИС эффективность последовательностей в среднем повысилась на 5...8%. Результаты моделирования были получены гораздо раньше, чем в обычной среде. Изначальное наличие последовательностей помогло быстрее достичь максимального покрытия. В целом при использовании улучшенной ПИС верификация интерфейса закончилась быстрее почти на 50%.

Таблица 3

Результаты сравнения улучшенной ПИС со стандартной средой

Номер последовательности	Среда с улучшенной ПИС (сек)	Среда Без ПИС (сек)	Эффективность (%)
1	71,8	76,2	+6,1
2	68,8	74,1	+7,7
3	66,5	72	+8,2
4	64	68,7	+7,3
5	72,2	76,1	+5,4
6	68,4	72,1	+5,4
7	74,8	78,6	+5,08
8	72,8	77,3	+6,1
9	42,3	45	+6,3

Третий разработанный метод позволяет динамически управлять функциональными точками и отключать те из них, которые уже выполнены. Это дает возможность снизить количество вызова функций покрытия, снижая этим время моделирования.

Важным компонентом верификации является решение, когда устройство можно считать проверенным. Имеется несколько видов покрытий: линий, конечные автоматы, условия и переключения. Эти виды покрытия автоматически генерирует симулятор и достижение высокого процента в этих видах покрытия требует больших временных затрат. Более быстро достигаемым видом покрытия является функциональное покрытие. Здесь во время планирования решаются функциональные точки (конфигурации). Если эти точки будут выполнены, то устройство можно считать проверенным (рис. 12). Проблема, которую можно решить данным методом, состоит в том, что эти функциональные точки выполняются более одного раза (рис. 13).

Cover Group Item	Score	Goal	Weight	At Least	Per Instance
uart_env_pkg_uart_fc_mgr_c_uart_cov_b	100.00%				
uart_env_pkg_uart_fc_mgr_c_uart_cov_b	100.00%	100%	1	1	1
uart_env_pkg_uart_fc_mgr_c_uart_cov_b	100.00%	100%	1	1	1
uart_env_pkg_uart_fc_mgr_c_uart_cov_b	100.00%	100%	1	1	1
uart_env_pkg_uart_fc_mgr_c_uart_cov_b	100.00%	100%	1	1	1
uart_env_pkg_uart_fc_mgr_c_uart_cov_b	100.00%	100%	1	1	1
uart_env_pkg_uart_fc_mgr_c_uart_cov_b	100.00%	100%	1	1	1
uart_env_pkg_uart_fc_mgr_c_uart_cov_b	100.00%	100%	1	1	1

Рис. 12. Пример завершенной верификации

Status	Bin Name	At Least	Size	Hit Count
✓	low_mid	1	1	137
✓	max	1	1	109
✓	mid_high	1	1	33
✓	zero	1	1	72

Рис. 13. Пример повторяющихся функциональных точек

Для решения этой проблемы в архитектуру тестируемой среды был добавлен блок динамического управления (рис. 14). Этот компонент проверяет функциональные группы и отключает те из них которые достигли 100% покрытия.



Рис. 14. Блок динамического управления в тестируемой среде

Незначительным недостатком этого метода является необходимость переписать функциональные точки в отдельные группы. Это условие является обязательным так как точки не могут проверять процент покрытия, а вот функциональные группы имеют соответствующие функции. Как только группа достигает 100% покрытия, блок динамического управления отключает ее.

Для проверки эффективности данного метода была разработана среда верификации интерфейса универсальный асинхронный приемопередатчик (УАПШ). После изучения документации УАПШ были разработаны функциональные точки. Затем, с помощью нескольких последовательностей было достигнуто полное функциональное покрытие.

Таблица 4

Результаты моделирования блока динамического управления

Функциональный компонент		Кол-во возможных случаев	Кол-во выполнения условий	Повторные выполнения	Разница выполнения условий (%)	Разница повторных условий (%)
Группа	Точка					
uart_cov_tx	Data	4	164	160	53	53
	Packet_width	4	147	143	58	58
	Parity_en	2	231	229	45	45
	Parity_odd_even	2	97	95	38	38
	Stop_bit_count	3	225	222	30	30
uart_cov_tx	Data	4	164	160	53	53
	Packet_width	4	147	143	58	58
	Parity_en	2	231	229	45	45
	Parity_odd_even	2	97	95	38	38
	Stop_bit_count	3	225	222	30	30

Из данных табл. 4 видно, что количество повторных выполнений условий функциональных точек снизилось на 30...50%. Это привело к снижению времени моделирования на 3...6%. Важно отметить, что самое большое отличие было зафиксировано для самых длинных последовательностей. Это показывает, что такой подход будет более эффективным в больших проектах и может быть невыгодным для маленьких (с точки зрения времени потраченного на полученный прирост эффективности). Недостатком метода является потребность менять код таким образом чтобы каждая функциональная группа состояла только из одной точки. Важно отметить, что этот метод зависит от случайности, а именно невозможно гарантировать минимальное число выполнения условий функциональной точки.

В третьей главе представлено программное средство для разработки среды верификации ИС, а также, дана оценка эффективности работы программы.

На основе доступных для создания среды верификации цифровых ИС и предлагаемых методик, описанных в предыдущей главе, было разработано программное обеспечение “TE Generator” для автоматизированного проектирования. Благодаря этому ПО можно автоматически генерировать среду верификации. ПО позволяет при помощи удобной графической среды моделировать разные среды верификации, добавлять любое количество компонентов (рис. 15) и модернизировать их, в соответствии с запросами пользователя. После моделирования ПО позволяет легко и быстро генерировать среду в удобном формате (рис. 16). Это дает возможность облегчить получение среды верификации с использованием методик УМВ и РМВ. Кроме этого, ПО сильно сокращает время, потраченное на создание среды верификации. Помимо создания среды данная ПО может генерировать улучшенную ПИС из любого компонента.

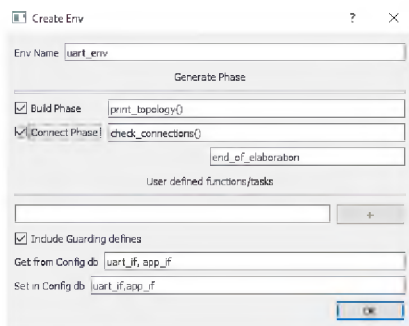


Рис. 15. Создание компонентов

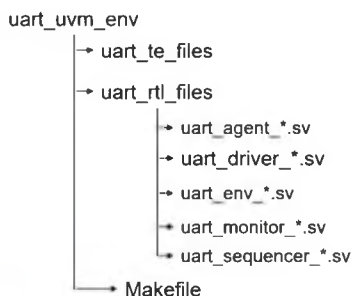


Рис. 16. Формат генерации среды

ПО “TE Generator” было создано с использованием языков программирования C++. Генерация компонентов была разработана с помощью языка Python, а графическую среду с помощью библиотек графического интерфейса QT. ПО “TE Generator” предназначено для операционной системы Windows.

В основном окне разработанного ПО (рис. 16) показана полностью готовая среда верификации для интерфейса УАПШ. Все представленные компоненты доступны для использования с панели инструментов, откуда можно добавить такие компоненты, как драйверы, генераторы, мониторы и т.д.

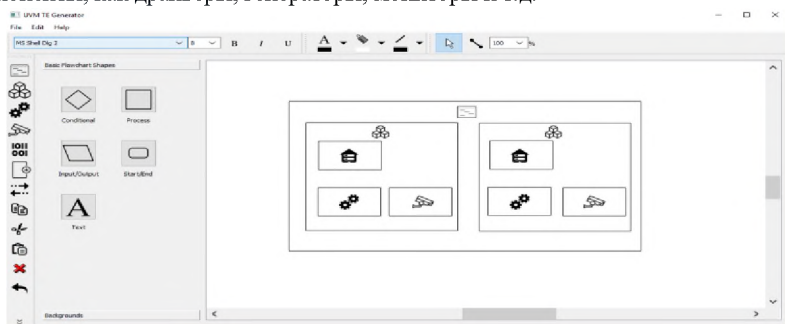


Рис. 17. Графический интерфейс ПО

Для оценки эффективности разработанного ПО “TE Generator” были созданы среда верификации УАПП и улучшенная ПИС УАПП. Для них был разработан план верификации. В первом случае была использована сгенерированная среда, а во втором - сгенерированная улучшенная ПИС была интегрирована в отдельную среду верификации.

Таблица 5

Сравнение ПО при использовании сгенерированной среды

Этап верификации	Планированное (день)	Традиционный метод	Сгенерированная среда	Разница (%)
Планировка верификации	1	1	1	0
Разработка среды	20	19	5	75
Разработка последовательностей	5	5	5	0
100% покрытие	10	10	10	0
Планировка верификации	36	35	21	42

Как видно из данных табл. 5, использование генерированной среды снижает время, потраченное на “Разработку среды”. На разработку среды при использовании ПО, по сравнению с традиционным методом, тратится почти на 75 % меньше времени, а в целом на верификацию 42%. Этот метод влияет только на разработку среды а метод улучшения ПИС имеет больше преимуществ.

Таблица 6

Результаты использования улучшенной ПИС

Этап Верификации	Планированное (день)	Традиционный метод	Верификация с улучшенной ПИС	Разница (%)
Планировка верификации	1	1	1	0
Разработка среды	20	19	6	70
Разработка последовательностей	5	5	3	40
100% покрытие	10	10	8	20
Планировка верификации	36	35	21	50

Результаты использования улучшенной ПИС (табл. 6) показывают, что этот метод не только влияет на время создания среды верификации, но также снижает время разработки последовательностей и получения максимального покрытия. Это объясняется тем, что блок начальной проверки уже имеет интегрированные последовательности. Как видно из результатов, использование ПИС снижает время разработки среды почти на 70%, время разработки последовательностей на 40% а для полного покрытия на 20%. В целом это сокращает время верификации почти на 50%. Недостатком обоих методов является отсутствие проверочных блоков в коде, что снижает точность среды в среднем на 10...15%. В случае, если инженер добавит проверочные блоки, ухудшение точности не будет.

ОСНОВНЫЕ ВЫВОДЫ ПО ДИССЕРТАЦИОННОЙ РАБОТЕ

1. Предложены способы разработки средств верификации проектов быстродействующих ИС, которые значительно улучшают их быстродействие, сокращают время выявления погрешности и продолжительность проектирования, повышают качество проверки. [1-9].
2. Разработан динамический метод управления проверочными точками средств верификации проектов быстродействующих цифровых ИС, который благодаря уменьшению числа повторяющихся измерений проверочных точек и аргументов, обеспечивает рост быстродействия последовательностей проверок на 3...5% за счет усложнения архитектуры среды [3].
3. Создан метод формирования среды верификации проектов быстродействующих цифровых ИС, который благодаря автоматической генерации компонентов обеспечивает уменьшение длительности процесса верификации в среднем на 42,7% за счет снижения точности работы функциональных блоков всего лишь на 10...15% [1].
4. Предложен метод улучшения характеристик проверочной интеллектуальной собственности (ПИС) средств верификации проектов быстродействующих цифровых ИС, который благодаря измененной архитектуре и внедрению последовательностей «начало-пауза-продолжение» обеспечивает повышение скорости процесса верификации почти на 50% и увеличение быстродействия проверочных последовательностей на 5...8%, за счет добавления компонентов в среду [4,5].
5. Разработан программный инструмент “TE Generator” для создание среды верификации проектов быстродействующих цифровых ИС, который имеет удобный для пользователя графический интерфейс и дает возможность создавать проверочную среду с использованием методологий “Универсальная методология верификации” или “Руководство по методике проверки”, а также генерировать среду улучшенной ПИС. Апробация программного обеспечения (ПО) в ряде настоящих проектов показала его высокую эффективность по сравнению с ручной разработкой среды верификации: за счет уменьшения точности проверочной среды всего на 10...15% обеспечивается сокращения срока проектирования в среднем на 40...70%. [6-9].

Основные результаты диссертации опубликованы в следующих работах:

1. Harutyunyan S.A., UVM Test Environment Generator. // Proceedings of the RA NAS and NPUA. Series of Technical Sciences. – 2022. – Vol. 72, No. 2. – P. 210–216.
2. Design and Verification of Novel Sync Cell /V. Melikyan, S. Harutyunyan, T. Kaplanyan, A. Kirakosyan, A. Momjyan, V. Amiryan // 2021 IEEE East-West Design and Test Symposium (EWDTS 2021): - Proceedings. – 2021. – P. 1–5. <https://doi.org/10.1109/EWDTS52692.2021.9580985>.
3. Melikyan V., Harutyunyan S., Kirakosyan A., Kaplanyan T. UVM Verification IP for AXI // 2021 IEEE East-West Design and Test Symposium (EWDTS 2021): - Proceedings. – 2021. – P. 1–4.
4. Configurable Verification IP for UART / S. Harutyunyan, A. Kirakosyan, H. Khachatryan, S. Harutyunyan, T. Kaplanyan // 2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO). – 2020. – P. 234–237.
5. Harutyunyan S., Kirakosyan A., Momjyan A., Kaplanyan T. Design And Verification Of Autoconfigurable UART Controller // 2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO). – 2020. – P. 347–350.
6. Melikyan V.SH., Harutyunyan S.A., Safaryan K.H., Implementation of D Flip-Flop For Ultra-Low Power Systems // Proceedings of the RA NAS and NPUA. Series of Technical Sciences – 2020. – Vol. 72, No. 2. – P. 210–216.
7. Harutyunyan S., Safaryan K., Bazikyan M., Aslikyan F. Design of Low Leakage SRAM Bitcell. // 2019 IEEE 39th International Conference on Electronics and Nanotechnology (ELNANO). – 2019. – P. 245–248.
8. Khachatryan. A., Harutyunyan H., Melikyan V., Harutyunyan S. A Novel Voltage Mode High Speed Sense Amplifier. // 2019 IEEE 39th International Conference on Electronics and Nanotechnology (ELNANO). – 2019. – P. 120–123.
9. Harutyunyan S., Safaryan K., Aslikyan F., Low Power Design of XOR Circuit // 2019 IEEE 39th International Conference on Electronics and Nanotechnology (ELNANO). – 2019. – P. 38–41.

ԱՄՓՈՓԱԳԻՐ

Ժամանակակից թվային ինտեգրալ սխեմաներում (ԻՄ) կիրառվող տրամաբանական տարրերի երկրաչափական չափերի փոքրացումը հնարավորություն է տալիս՝ ունենալու էլ ավելի շատ ֆունկցիոնալ բաղադրիչներ: ԻՄ-ի այդպիսի զարգացման պայմաններում պետք է հաշվի առնվի նաև նախագծերի ստուգման վրա ծախսվող ժամանակը: Դա պայմանավորված է նրանով, որ յուրաքանչյուր ֆունկցիոնալ բաղադրիչ պահանջում է առանձին ստուգման գործընթաց: Նախագիծը, ունենալով մեծ քանակությամբ բաղադրիչներ, կպահանջի շատ երկար ժամանակ՝ ստուգման գործընթացը կազմակերպելու համար: Դա կհանգեցնի կամ շուկայում մրցակցության կորստի, կամ արտադրանքի որակի նվազման: Ներկայացված խնդիրները լուծելու նպատակով անհրաժեշտ է կիրառել թվային ԻՄ-երի նախագծերի ստուգման արագագործ միջոցներ:

Վերջին տարիներին նախագծերի չափերի աճման տեմպի հետևանքով, հաճախ նախագծի տևողության 70-80%-ը զբաղեցնում է դրա ստուգման փուլը: Առավելագույն որակ ապահովելու համար անհրաժեշտ է ստուգել հնարավոր բոլոր մուտքային համադրումները, հակառակ դեպքում՝ կստացվեն ծածկույթի բացթողումներ, որոնք կարող են հանգեցնել արտադրանքի որակի և հուսալիության հետ կապված խնդիրների: Դրա հետևանքով ծախսվում է հսկայական ժամանակ՝ ստուգման գործընթացը ծածկույթների միջոցով կազմակերպելու համար: Ծածկույթների առավելագույն արժեքին հասնելը չափազանց երկար գործընթաց է: Անհրաժեշտ է ստեղծել հատուկ ստուգման հաջորդականություններ և մոդելավորել դրանք: Ծածկույթի որոշ տեսակներ, առավելագույն արժեք ունենալու համար, պահանջում են նախագծի յուրաքանչյուր տողի մոդելավորում: Շուկայում ներդրվում են որակի ստանդարտներ, որոնք պարտադրում են ունենալ ծածկույթների չափազանց մեծ՝ 95-98% արժեք, որը պետք է ապահովվի շատ արագ՝ մրցակցությունը չկորցնելու համար: Այդպիսի մարտահրավերների պարագայում առկա մեթոդները չեն բավարարում առաջադրված պահանջները և անհրաժեշտ է մշակել նոր, բարձր արագագործությամբ ստուգման միջոցներ:

Ստուգման գործընթացը կազմակերպելու համար առկա են մի քանի մեթոդներ: Շուկայում առաջատար կազմակերպություններից յուրաքանչյուրն առաջարկում է իր մեթոդը, սակայն դրանց ներդրումն ու կիրառումը նույնպես չափազանց ժամանակատար են:

Ստուգման գործընթացը սկսվում է պլանի կազմումից, որն ընգրկում է միջավայրի ստեղծումը, հաջորդականությունների ձևավորումը և մոդելավորումը, առավելագույն ծածկույթին հասնելու ժամանակը: Նշված յուրաքանչյուր բաղադրիչ զբաղեցնում է զգալի ժամանակ, սակայն ամենաժամանակատարը առավելագույն ծածկույթի ապահովումն է: Ունենալ առավելագույն ծածկույթ նշանակում է մոդելավորել սարքի ռեգիստրափականային մակարդակով նկարագրության յուրաքանչյուր տող, պայման,

վերջավոր ավտոմատի վիճակ, փոխանջատում: Դա շատ երկար ժամանակ է պահանջում, ուստի հաճախ սահմանափակվում են ֆունկցիոնալ ծածկույթի կիրառմամբ, երբ ստուգվում են մասնագետի կողմից նախապես որոշված պայմանները: Այդ մեթոդը, ունենալով արդյունավետություն արագագործության տեսանկյունից, կարող է ունենալ բացթողումներ ոչ բավարար պայմանների դեպքում, ինչը կարող է հանգեցնել սարքի սխալ աշխատանքին: Նշվածներից պարզ է դառնում ստուգման գործընթացի հսկայական տևողության անհրաժեշտությունը: Սակայն այդքան մեծ ժամանակի ծախսը միշտ չէ որ ապահովում է ցանկալի արդյունքը: Ժամանակակից նախագծերի մեծ մասը չի ավարտվում նախապես որոշված ժամանակահատվածում: Դա ստեղծում է բարձր արագագործությամբ ստուգման միջոցների մշակման անհրաժեշտություն:

Առաջարկվել են բարձր արագագործությամբ թվային ինտեգրալ սխեմաների նախագծերի ստուգման միջոցների մշակման եղանակներ, որոնք էապես բարելավում են դրանց արագագործությունը, սխալների դուրսբերման ժամանակը, ստուգման որակը և նախագծման տևողությունը [1-9]:

Մշակվել է ստուգման կետերի դինամիկ ղեկավարման մեթոդ, որը չափման կետերի և պնդումների բազմակի չափումների նվազեցման շնորհիվ ապահովում է ստուգման հաջորդականությունների արագագործության 3-5% աճ միջավայրի ճարտարապետության բարդացման հաշվին [3]:

Առաջարկվել է ստուգման միջավայրի գեներացման մեթոդ, որը, դրա բաղադրիչների ավտոմատ գեներացման շնորհիվ, ապահովում է ստուգման գործընթացի տևողության փոքրացում 42%-ով՝ ի հաշիվ ֆունկցիոնալ բլոկների աշխատանքային ճշտության ընդամենը 10-15% նվազման [1]:

Ստեղծվել է ստուգման մտավոր սեփականության բնութագրերի բարելավման մեթոդ, որը, դրա ճարտարապետության փոփոխության և «սկիզբ-դադար-շարունակություն» հաջորդականությունների ներդրման շնորհիվ, ապահովում է ստուգման գործընթացի մոտավորապես 50% և հաջորդականությունների 5-8% արագագործության աճ՝ միջավայրում հավելյալ բաղադրիչի ավելացման հաշվին [4,5]:

Մշակվել է բարձր արագագործությամբ թվային ինտեգրալ սխեմաների ստուգման միջոցների ստեղծման TE Generator ծրագրային գործիքը, որն օժտված է օգտագործողի համար հարմարավետ գրաֆիկական ինտերֆեյսով և հնարավորություն է ընձեռում՝ ստեղծել «ստուգման համապիտանի մեթոդոլոգիա» կամ «Ստուգման Մեթոդոլոգիայի ձեռնարկ» մեթոդոլոգիաներով ստուգման միջավայրեր, ինչպես նաև գեներացնել բարելավված ստուգման մտավոր սեփականության միջավայր: Վերջինիս փորձարկումը մի շարք իրական նախագծերում ցույց է տվել ձեռքով մշակված ստուգման միջավայրերի համեմատ, դրա բարձր արդյունավետությունը. ստացված միջավայրի ճշտության ընդամենը 10-15%-ով նվազման պարագայում ապահովվում է նախագծման ժամկետի միջինում 40-70% կրճատում [1,2,6-9]:

DEVELOPMENT OF VERIFICATION METHODS FOR HIGH PERFORMANCE DIGITAL INTEGRATED CIRCUITS

SUMMARY

The development of the technology allows more functional components in the integrated circuits. Higher number of components means more verification that needs to be done. This arises problems from time consumption perspective. Sometimes the verification is taking as much as 70-80% of the entire project duration. Current market demands the projects to be completed as fast as possible. Taking into account increasing number of components, hence bigger verification duration, the need of faster means of verification becomes more vital.

In recent years, various methodologies have been introduced to decrease the time spent on verification. Those methods were mainly developed by industry leading companies. Each one of them introduced special methodology which was giving boost in verification speed but it was hard to implement and had other shortcomings. The verification process itself has several stages. First of all verification plan is made, after which verification environment is developed. When the environment is ready test sequences are developed and simulated. The last component needed is the full coverage. All these items are time consuming but biggest challenge is the full coverage. Full coverage means: all RTL lines have been exercised, all conditions checked, all transitions simulated and all the FSM states covered. That takes huge amount of time and mostly avoided to meet challenging market deadlines. Not having the full coverage on the other hand means there could be potential issues missed which at the end can result in post production issues. To mitigate this, functional coverage is used which uses verification engineer defined functional bins to tell, if the verification of the design is completed or not.

The above mentioned explain the long duration of the verification, but even if such huge time is spent on verification it doesn't guarantee the result. Only about 20-25% of the projects are completed on time, rest are postponed by up to 50% of the initial planned time. In order to meet the current market challenges several high speed solutions were proposed.

The principles for the development of verification methods for high performance digital integrated circuits has been proposed. These principles will allow to significantly reduce the performance of the design, the debug time, quality of the verification and the duration of the development [1-9]

A method to dynamically control functional bins of the high performance verification methods for digital integrated circuits have been developed, which, due to reduction of multiple duplicate measurements of assertions and functional bins, provides about 3-5% increased productivity of test sequences, at the expense of increased difficulty of the architecture [3].

A method to form the verification environment for high performance digital integrated circuit projects has been developed, which due to automated generation of environmental components, lowers the verification time in average by 42,7%, at the expense of 10-15% of decreased accuracy of the functional blocks. [1].

A method to improve the characteristics of the verification intellectual properties for high performance digital integrated circuit projects has been developed, which due to improved architecture and usage of “start-stop-continue” sequences, provides reduction of time required for verification by 50% and for test sequences by 6-8%, at the expense of additional components in the environment. [4,5].

A “TE Generator” tool for development of high performance verification methods for digital integrated circuits has been created, which has user-friendly graphical user interface and allows to create test environments with methodologies such as “Universal Verification Methodology” or “Verification Methodology Manual” as well as generate improved verification intellectual property environments. The tests of the latter in several real projects compared to hand created test environment shows its effectiveness, at the cost of 10-15% decreased accuracy, the verification time has been decreased by in average of 40-70%.

