



## МАТЕМАТИКА И ИНФОРМАТИКА

УДК 004.052.42

Т.С. Титовская, О.В. Непомнящий,  
А.В. Леонова, А.А. Комаров

### ФОРМАЛЬНАЯ ВЕРИФИКАЦИЯ ПРИ ПРОЕКТИРОВАНИИ СВЕРХБОЛЬШИХ ИНТЕГРАЛЬНЫХ СХЕМ

Рассмотрены основные маршруты высокоуровневого синтеза вентильных описаний СБИС. Определены проблемы верификации проекта на функциональном уровне, присущие традиционным методам проектирования. Изложены основные положения разрабатываемой технологии архитектурно-независимого представления СБИС на основе функционально-поточковой парадигмы параллельного программирования. Предложен подход к формальной верификации проекта при высокоуровневом синтезе.

**Ключевые слова:** верификация, сверхбольшие интегральные схемы (СБИС), функциональное программирование, параллельные вычисления.

T.S. Titovskaya, O.V. Nepomnyashchiy,  
A.V. Leonova, A.A. Komarov

### FORMAL VERIFICATION IN VERY LARGE-SCALE INTEGRATION DESIGNING

The main routes of the high-level synthesis of the VLSI gated descriptions are considered. The project verification problems on the functional level, inherent in traditional methods of design are defined. The basic provisions of the developed architecture-independent technology for VLSI representation on the basis of the functional-flow paradigm of parallel programming are presented. The approach to the project formal verification in the high level synthesis is offered.

**Key words:** verification, very large-scale integration (VLSI), functional programming, parallel computing.

**Введение.** Степень интеграции современных сверхбольших интегральных схем (СБИС) растет экспоненциально. На сегодняшний день переход к субмикронным технологическим нормам позволяет размещать на кристалле практически любой по сложности проект. Тем не менее имеется другой сдерживающий фактор – проблема функциональной верификации СБИС [1].

По последним данным, примерно половина всего инженерного состава, работающего над крупными проектами, занята верификацией, а временные затраты на нее в общем цикле проектирования выглядят еще более впечатляюще – более 60 % [2]. Такое положение обуславливает поиск принципиально новых решений, разработку передовых методов и технологий верификации проекта на протяжении всего технологического цикла.

**Цель исследования.** Разработка новых подходов к архитектурно-независимому описанию и функциональной верификации СБИС.

**Задачи исследования:** изучение и разработка перспективных методов проектирования и верификации высокоуровневых представлений однокристалльных систем, а также разработка алгоритмов и инструментальных средств формальной верификации функционально-поточковых представлений СБИС.

**Методики исследования.** Теоретической и методологической основой исследований послужили разработки отечественных и зарубежных ученых в области проектирования и верификации СБИС. Основными информационными источниками явились материалы научно-практических конференций, сборники научных статей, а также ранее выполненные научно-исследовательские разработки и публикации по исследуемой тематике.

Традиционный маршрут нисходящего проектирования подразумевает изначальное формирование спецификаций – временных и других ограничений для каждой составной части проекта, поэтапное разбиение системы сначала на крупные, а затем на меньшие блоки, впоследствии реализуемые на уровне RTL-описания (Register Transfer Level).

Основные недостатки такого подхода кроются в существенной разнице между уровнями абстракции при функциональном и RTL-описаниях проекта. При разработке системы на верхних уровнях абстракции не

учитывается целый ряд параметров функционирования, (например, временные ограничения), в то время как при описании на HDL-языке они выходят на первый план.

При таком подходе каждый блок проектируется и верифицируется отдельно, и только после объединения в более крупные блоки происходит совместная верификация. Таким образом, более мелкие блоки приходится верифицировать как минимум дважды. Сначала функционирование блока тестируется на соответствие его спецификации, а затем на соответствие общим требованиям в составе интегрированной системы. При этом имеется высокая вероятность обнаружения ошибки на одном из последних этапов интеграции, в особенности – на уровне системной верификации, и, как следствие, возврат к началу проектирования.

Дальнейшее развитие традиционного метода на основе добавления функциональных моделей на верхнем уровне абстракции или попытки их анализа на RTL-уровне привели к появлению понятия проектирования и формальной верификации на абстрактном – системном уровне (Electronic System Level, ESL), которое активно развивается с 2004 года. При таком подходе на верхнем уровне абстракции к традиционному этапу описания и формирования пакета спецификаций добавляют два дополнительных уровня: уровень сообщений (Message Level) и уровень передач (Transaction Level). При этом спецификация системы описывается на уровне объектных диаграмм.

В отличие от традиционного маршрута, на начальном этапе ESL-проектирования формулируется задача на проектирование, описывается состав системы в виде функциональных блоков и связей между ними, а также происходит отладка их взаимодействия с целью проверки разработанного функционального состава на соответствие поставленной задаче. В этом случае разработка, отладка и верификация алгоритмов, решающих заданную прикладную задачу, полностью отделены от последующего процесса переложения этих алгоритмов на архитектуру СБИС.

После создания модели происходит формирование архитектуры системы. В ходе этого процесса сформированные на предыдущем этапе блоки описываются на стандартных языках программирования. Для разработанных блоков генерируются тестовые воздействия, фрагменты отладочного программного обеспечения и тесты для будущих аппаратных модулей. На основании тестов и разработанного программного обеспечения происходит детальная верификация полученной программной модели.

При обнаружении в ходе верификации ошибок или несоответствия поставленной задаче происходит возврат на первоначальный уровень разработки, декомпозиция или разработка новых блоков спецификации. Этот процесс повторяется до полного устранения ошибок и несоответствий.

Такой подход не позволяет однозначно утверждать, что все возможные варианты реализации проекта будут рассмотрены. Однако отладка на самом верхнем уровне позволяет существенно сэкономить общее время проектирования, хотя и требует высокой квалификации проблемно-ориентированных разработчиков.

Несмотря на преимущества формальной верификации при ESL перед традиционным подходом, нерешенным остается ряд проблем, характерных для обеих методик. В частности, для получения оптимального по соотношению *производительность/занимаемое на кристалле место* требуется выполнить анализ множества вариантов реализации проекта. При этом на каждой стадии проектирования требуется обязательная верификация системы, то есть необходима разработка и генерация пакета тестов для каждого этапа. Кроме того, в процессе тестирования возникают сложности с внесением изменений. На алгоритмическом уровне модификации просты, но при переходе к RTL-описанию вновь может потребоваться перекомпоновка всего проекта и его повторная верификация.

На основании вышеизложенного отметим, что при проектировании сложных систем требуется новый, архитектурно-независимый подход, с одной стороны, обеспечивающий максимальное абстрагирование исходных алгоритмов от архитектуры целевого кристалла, с другой стороны, реализующий механизм перехода на RTL-уровень с параллельной верификацией при проектировании без возврата к предыдущим уровням создания проекта.

Необходим такой способ представления алгоритмов на самом верхнем уровне иерархии, который при последующем нисходящем проектировании обеспечивал бы сохранение параллелизма, задаваемого на самом верхнем уровне, и допускал бы сквозную верификацию без возврата к предыдущим уровням иерархии.

Таким образом, для эффективного решения задач высокоуровневого синтеза СБИС требуется использование модели вычислений с максимальным параллелизмом, которая позволяет описывать алгоритмы функционирования СБИС без привязки к конечной реализации. В этом случае требуется переход от традиционного представления к использованию параллелизма на уровне операций. Следовательно, необходимо использовать соответствующую модель вычислений и разработанный на ее основе язык программирования. В работе [3] предложены функционально-потокосная модель параллельных вычислений и язык программирования Пифагор, использование которых позволяет по-иному взглянуть на проектирование сверхбольших интегральных схем.

Рассмотрим в общем виде модель цифровой системы. Любая такая система является набором блоков, соединенных между собой. Связи между блоками можно разделить на два типа: линии данных и управления. Каждый блок может быть как элементарным (выполняющим элементарную функцию – арифметическую, логическую и т.д.), так и составным. Цифровые схемы работают в синхронном потоковом режиме, то есть в зависимости от состояния сигналов управления блок обрабатывает входные данные. Следовательно, цифровая схема реализует граф потока данных, где узлами являются блоки, а ребрами – сигналы данных и управления. Блоки в цифровой схеме работают параллельно и квазинеzáвисимо друг от друга. Зависимость между блоками определяется только сигналами управления, которые определяют готовность данных на его входах.

Аналогом графа с блоками обработки данных, сигналами управления и схемами их формирования являются информационный и управляющий графы функционально-поточковой модели соответственно. Процесс перехода от этой модели к описанию цифровой системы будет представлять собой анализ информационного графа, в результате которого получается ряд промежуточных представлений согласно фазам обработки потока данных. Одно из таких представлений будет использоваться для последующего синтеза и многокритериального анализа цифровой системы.

Для достижения максимальной эффективности при распараллеливании, с учетом ресурсных ограничений кристалла, на этапе построения промежуточного слоя происходит оптимизация и формальная верификация фаз.

На этапе трансляции в RTL-представление происходит построение модели управляющего графа на базе процессорных ядер с заданной программой управления, в то время как информационный граф реализуется в виде конечных аппаратных примитивов. По окончании построения происходит генерация полученных моделей в HDL-описание цифровой схемы.

Применение модели на основе графа потока данных позволяет реализовать оптимизацию вычислений с использованием алгоритмов оптимизации графов. Это позволяет оптимизировать архитектуру СБИС на этапе формального описания схемы, до перехода к архитектурному описанию системы, что не выполняется ни в одном из существующих методов разработки.

Таким образом, процесс формальной верификации СБИС при архитектурно-независимом описании на основе функционально-поточковой модели вычислений сводится к сравнению информационных и управляющих графов функционально-поточковой модели: одни содержат сведения о разрабатываемой системе в условиях неограниченного параллелизма, вторые – ориентированы на имеющиеся вычислительные ресурсы. Для обеспечения корректного доказательства логической эквивалентности этих графов могут применяться формальные математические методы.

**Выводы.** В отличие от традиционных маршрутов высокоуровневого синтеза, разрабатываемая технология и инструментальные средства формальной верификации позволяют оперировать не на уровне готовых аппаратных платформ или ранее разработанных и верифицированных блоков для однокристалльных систем, а на уровне принципов системной организации вычислительного процесса, с переносом полученной модели на целевой кристалл.

При архитектурно-независимом представлении СБИС на функционально-поточковом языке параллельного программирования верификация и оптимизация архитектуры СБИС обеспечиваются в течение всего цикла проектирования. Система тестируется автоматически при создании каждого промежуточного представления на верхних уровнях иерархии.

Таким образом, использование методов формальной верификации и поддержки параллелизма на уровне операций и параллельной потоковой модели на всех стадиях процесса высокоуровневого проектирования СБИС позволяет выйти на качественно новый уровень проектирования однокристалльных систем.

### Литература

1. *Непомнящий О.В., Алекминский С.Ю.* Проблемы верификации проекта при сквозном проектировании вычислительных систем на кристалле – нано- и микросистемная техника. – М.: Новые технологии, 2010. – № 9 (122). – С. 4–7.
2. *Немудров В., Мартин Г.* Системы-на-кристалле. Проектирование и развитие. – М.: Техносфера, 2004.
3. *Легалов А.И.* Функциональный язык для создания архитектурно-независимых параллельных программ // Вычислительные технологии. – 2005. – № 1 (10). – С. 71–89.

