

Применение поведенческих моделей при проектировании систем на кристалле

77-48211/479481

09, сентябрь 2012

Котельницкий А. В., Власов А. И.

УДК. 519.876.5

Россия, МГТУ им. Н.Э. Баумана

lex.skorpio@gmail.com

Введение

В современных условиях, критически важным является время проектирования электронного устройства. Одним из способов уменьшения времени разработки, является использование поведенческих моделей. Такая модель, описывает функционал проектируемого устройства в целом, задает соответствие входных и выходных сигналов системы, имитируя ее работу. При этом модель не учитывает внутренней логики устройства и не может быть использована для дальнейшего синтеза принципиальной схемы. Такая структура модели обусловлена основным ее применением – предварительным моделированием работы устройства для анализа разработанной архитектуры и принятых принципов работы. При системном подходе, поведенческая модель может быть построена на основе визуальных объектных языков моделирования (UML) и реализована впоследствии на языках программирования высокого уровня. Обычно для этой цели выбирается язык C++ с использованием библиотеки SystemC [2]. Это обусловлено простотой разработки модели относительно языков описания аппаратуры, высокой скоростью компиляции и выполнения кодов [1, 2]. Библиотека SystemC необходима для моделирования работы сигналов, и параллельного выполнения команд.

Поведенческие модели систем на кристалле создаются для решения различных задач – для функционального анализа предполагаемой архитектуры устройства [1, 3], верификации [2, 3] и отладки разрабатываемого программного обеспечения (драйверов) [3]. Основной задачей, которую необходимо решить, прежде чем приступить к разработке поведенческой модели, является ее детализация. С увеличением детализации растет время на ее разработку, вносить изменения в более подробную модель сложнее. Однако при уменьшении детализации применимость модели в процессе проектирования электронного устройства падает - использовать простую модель можно только для конкретных локальных задач.

Применение поведенческих моделей для описания систем на кристалле было рассмотрено в работах [2, 3]. Так, в работе [2] предложено компромиссное решение, при котором разрабатывается функциональная модель без учета временных задержек с дополнительным блоком-арбитром. Арбитр распределяет эти задержки на входные и выходные сигналы, имитируя реальную работу устройства. Данное решение позволит использовать разрабатываемую модель, как для функционального анализа, так и для верификации, путем включения или отключения блока-арбитра. Однако подобную модель нельзя использовать с целью дальнейшего уточнения, верификации и анализа входящих в нее блоков.

В работе [3] поведенческие модели, используемые для различных задач, рассматриваются как независимые друг от друга. Данный метод повышает эффективность использования моделей для локальных задач, но требует большого количества времени для их разработки.

Таким образом, для оптимизации проектирования электронного устройства, необходимо иметь в виду все возможные применения поведенческой модели при ее разработке. В работе рассматриваются основные применения моделей в рамках проектирования систем на кристалле. Отдельно рассмотрено применение поведенческих моделей при проектировании суперкомпьютерной сети «Ангара».

1 Маршрут проектирования систем на кристалле с использованием поведенческих моделей

Применение поведенческих моделей при проектировании систем на кристалле позволяет уменьшить количество возможных ошибок на самых ранних этапах проектирования, упростить верификацию логических блоков и тем самым ускорить процессы разработки электронных средств [1].

На рисунке 1 изображена диаграмма, описывающая основные этапы первой ступени разработки системы на кристалле (front end of line) с использованием поведенческой модели.

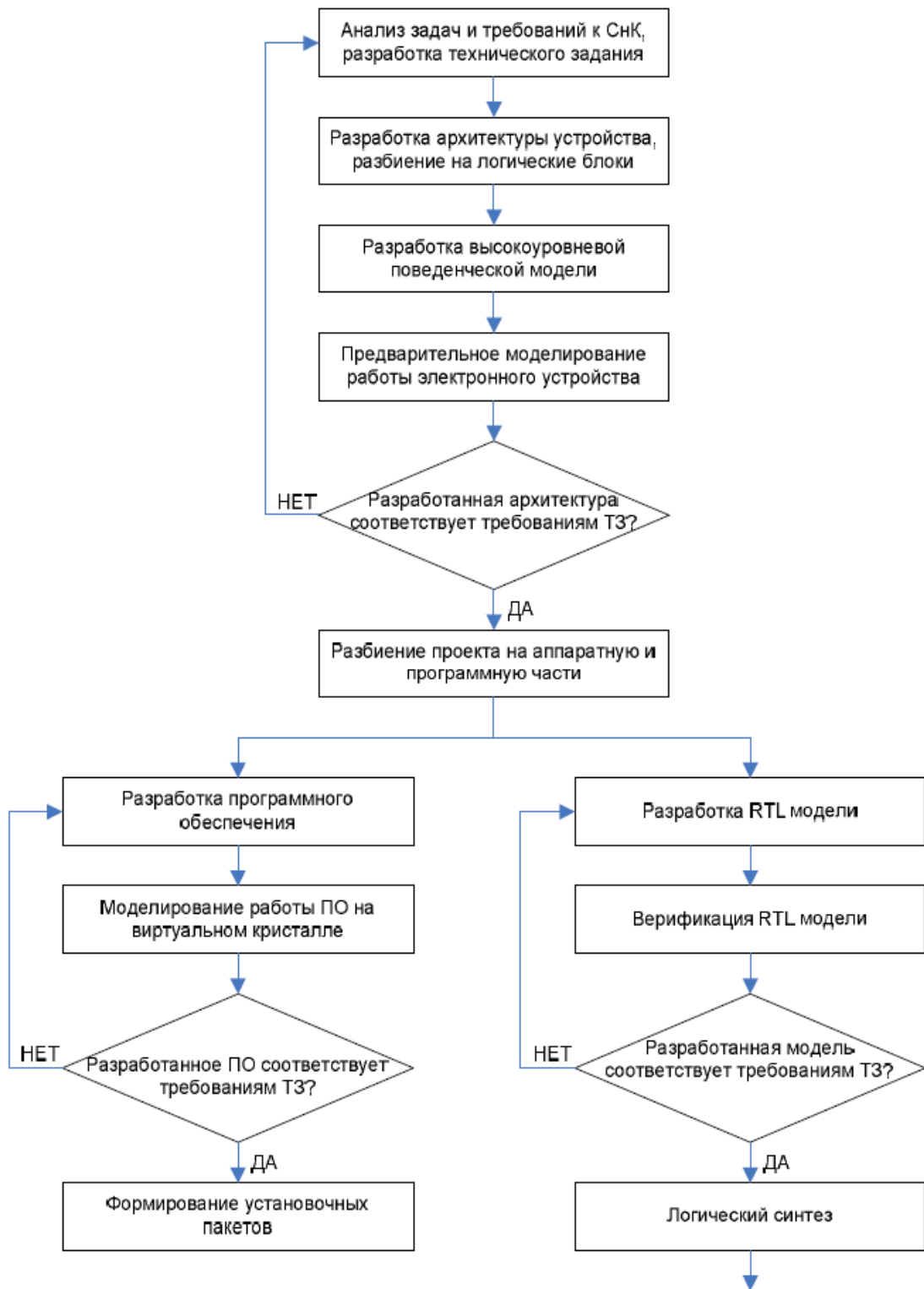


Рисунок 1 – Основные этапы разработки систем на кристалле

Процесс начинается с разработки технических требований, их последующего анализа и разбиения устройства на логические блоки. Затем

проводится функциональная проверка технических требований и анализ разработанной архитектуры устройства. С этой целью разрабатываются поведенческие модели логических блоков, и проводится их моделирование. После функциональной проверки технических требований в полученной модели выделяются подзадачи, выполняемые аппаратными или программными средствами. В результате проект разбивается на две части: программную и аппаратную. В дальнейшем разработка программного обеспечения ведется параллельно основному проекту.

При отладке программного обеспечения разработанная ранее поведенческая модель может быть использована в качестве виртуального кристалла [3]. Обычно для этих целей используют ПЛИС или прототип кристалла. Однако это возможно только на последних стадиях проектирования, когда в распоряжении разработчиков уже есть отлаженная RTL-модель (Register-Transfer Level), из которой можно было бы сгенерировать код для программирования ПЛИС. Для сокращения сроков выпуска изделия на первых этапах проектирования систем на кристалле (СнК) в качестве виртуального кристалла используют поведенческую модель. В этом случае ПО разрабатывается поверх модели, а при симуляции коды объединяются и компилируются вместе. Отладка при помощи виртуального кристалла может также применяться при ограниченном финансировании проекта, поскольку данный метод не предполагает дополнительных затрат на покупку ПЛИС.

На следующем этапе аппаратной части проекта создается RTL модель, описывающая схему на уровне регистров. Для разработки используется один из языков описания аппаратуры (VHDL, Verilog). Затем по RTL описанию с помощью программы логического синтеза формируется список цепей и топология схемы.

2 Применение поведенческих моделей при верификации

Почти одновременно с началом создания RTL модели начинается процесс ее верификации. Процесс начинается с разработки драйвера – программного модуля, который подает тестовые входные данные на RTL модель. Затем необходимо проверить выходные данные модели на соответствие ТЗ. В отсутствие поведенческой модели проверка осуществляется с помощью многочисленных локальных тестов (рис.2). Однако такая методика требует больших временных затрат на их программирование.

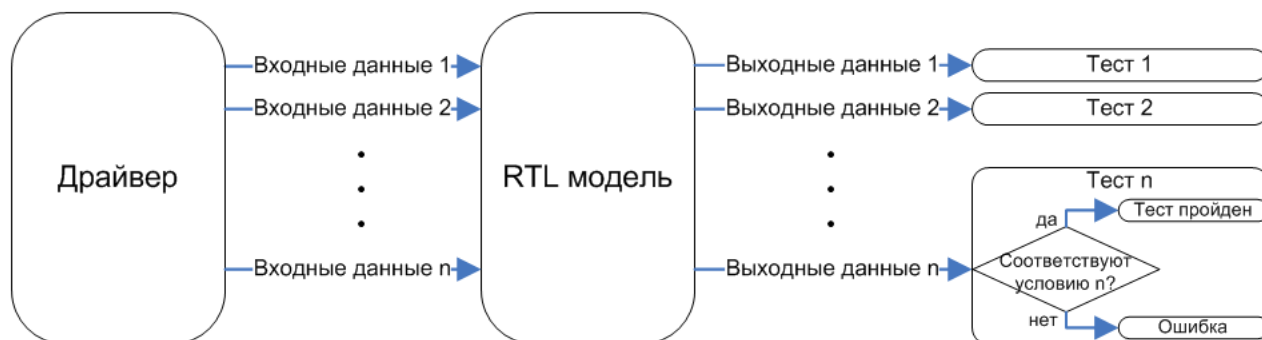


Рисунок 2 – Верификация RTL модели в отсутствие поведенческой модели

Наличие разработанной поведенческой модели позволяет значительно упростить процесс верификации. В данном случае, поведенческая модель используется в качестве эталонной (golden model) [3,6]: поскольку к моменту верификации RTL модели, поведенческая модель уже отлажена, ее работа считается правильной и берется за образец. Схема процесса верификации RTL модели с использованием поведенческой модели показана на рисунке 3.

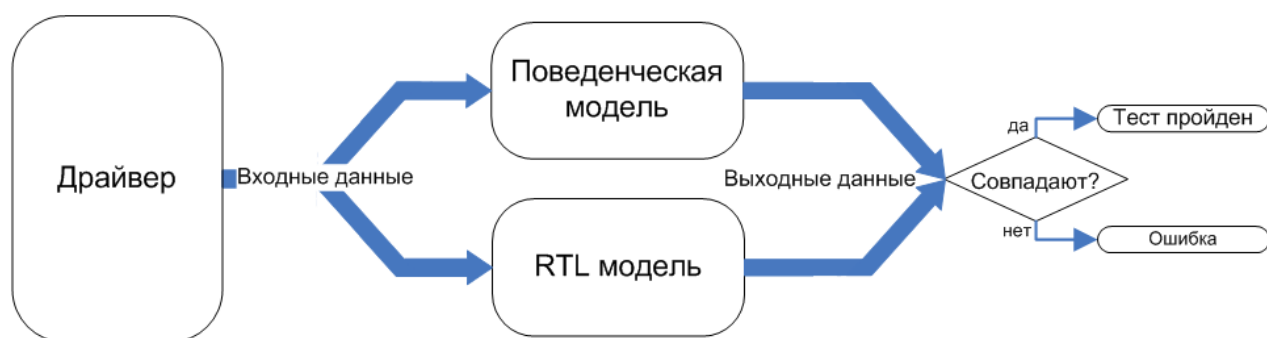


Рисунок 3 – Верификация RTL модели с использованием поведенческой модели

Таким образом, процесс верификации сводится к подаче тестовых данных на вход RTL и эталонных моделей и последующему сравнению выходных сигналов. Если результат работы двух моделей идентичен, делается вывод о правильной работе RTL кода.

4 Применение поведенческой модели при проектировании маршрутизатора суперкомпьютерной сети «Ангара»

Поведенческая модель при проектировании маршрутизатора суперкомпьютерной сети «Ангара» применяется для моделирования работы коммуникационной сети. Моделирование сети проводится для отладки принятых принципов маршрутизации, анализа производительности проектируемой сети и верификации.

«Ангара», как и большинство современных вычислительных комплексов, строится по принципу разделяемой памяти с общим адресным пространством, что приводит к повышению межзлогового трафика сети [5]. В таких условиях коммуникационная сеть является одной из важнейших частей суперкомпьютера, определяющей его производительность и

масштабируемость. Именно поэтому, особое внимание разработчиков уделяется разработке маршрутизатора и отладке принятых принципов маршрутизации данных по сети.

Коммуникационная сеть «Ангара» является сетью с топологией «Многомерный Тор» [4], в каждом узле которой находится маршрутизатор. Таким образом, для моделирования работы сети необходимо собрать в сеть соответствующее количество копий моделей маршрутизатора. Каждая копия, в таком случае, представляет собой узел со встроенным маршрутизатором. Моделирование работы подобной модели сети требует больших вычислительных мощностей. При использовании в узлах RTL моделей, моделирование нескольких секунд работы сети из 9 узлов может достигать месяца. В этих условиях скорость выполнения кода модели является критически важным показателем. Для увеличения скорости моделирования RTL модель заменяется поведенческой. При этом скорость моделирования увеличивается многократно, за счет низкой детализации поведенческой модели и высокой скоростью выполнения C++ относительно языков описания аппаратуры [2]. Для оценки работы разработанной RTL модели и ее верификации, только в одном из узлов поведенческая модель замещается соответствующей RTL. На рисунке 4 изображена схема описанной выше модели коммуникационной сети.

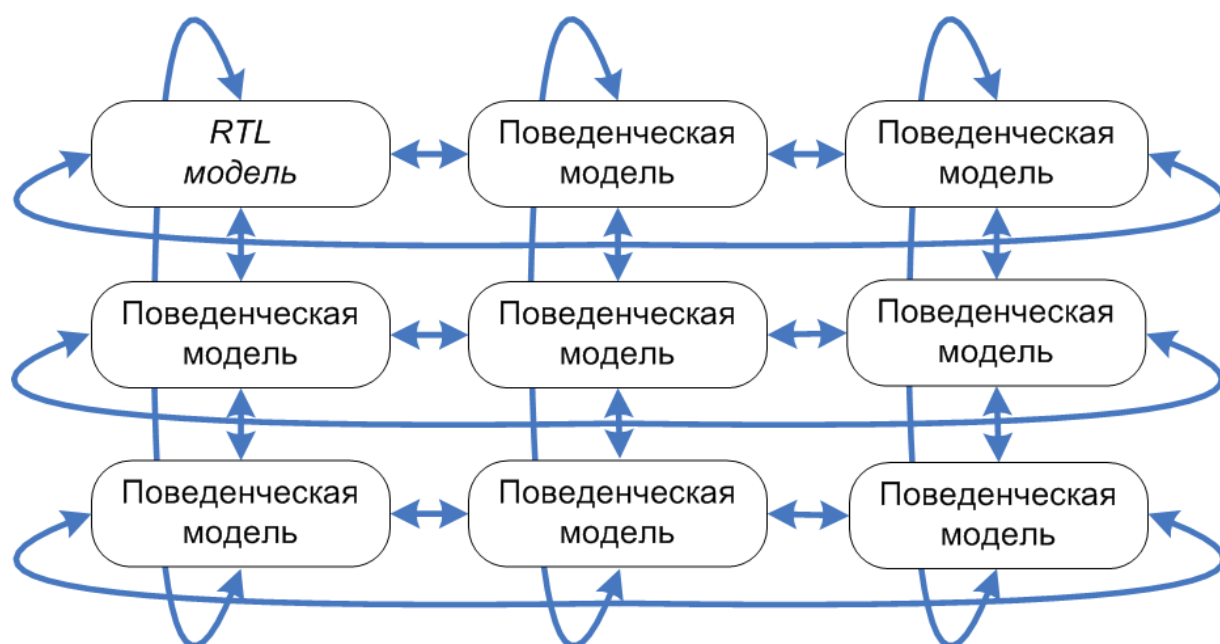


Рисунок 4 – Схема модели коммуникационной сети «Ангара»

Для анализа работы RTL модели, при помощи специального программного обеспечения собирается статистика, о том, какие из блоков являются наиболее загруженными во время работы сети, где происходят наибольшие задержки при передаче данных. На основе полученной информации делается вывод о внесении изменений в проектируемый маршрутизатор.

Верификация модели осуществляется при помощи контроля входных и выходных данных на каждом узле. В случае возникновения непредвиденных ошибок в работе сети, делается вывод о неправильной работе RTL модели.

Данный пример наглядно иллюстрирует возможность применения поведенческих моделей для функциональной оценки работы СнК и верификации устройства.

Список литературы

1. Шагурин И. В., Канышев В. А. Применение языка SystemC и средств разработки на его основе для проектирования «Систем на кристалле» / А. С. Кочнев // Инженерная практика. – 2006. – Т. 32. – №9. – С. 23–32.
2. Слепов А.Б. Разработка потактовой поведенческой модели системы на кристалле на языке C++ // Проблемы разработки перспективных микро- и нанoeлектронных систем. Сборник трудов / под общ. ред. академика А.Л.Стемпковского. - М.: ИППМ РАН, 2010. С. 450-453.
3. S. Velusamy, A. Mehta. Unified Modeling in C++ and SystemC // Wipro Technologies/ Bangalore.
4. Макагон Д., Сыромятников Е. Сети для суперкомпьютеров// Открытые системы. - №7. - 2011.
5. Слуцкий А., Эйсымонт Л. Российский суперкомпьютер с глобально адресуемой памятью // Открытые системы. - №9. – 2007.
6. Стешенко В., Руткевич А., Гладкова Е., Шишкин Г., Воронков Д. Проектирование СБИС типа «Система на кристалле». Маршрут проектирования. Синтез схемы // Электронные компоненты. - №1. – 2009.