

ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ СИСТЕМ НА КРИСТАЛЛЕ

Хабаров В.А., Бабаков А.В., Дронов Е.А.

Научный руководитель – канд. техн. наук, доцент Непомнящий О.В.

Сибирский федеральный университет

Под системой на кристалле SoC (System on Chip) подразумевают вычислительный комплекс, реализованный в интегральном исполнении, то есть в одной микросхеме. Здесь на одной кремниевой подложке располагаются высокопроизводительный процессор или несколько процессоров, сопроцессоры математической обработки данных и цифровой обработки сигналов, дополнительные модули памяти, набор периферийных устройств (контроллеров) и т.д. Такая организация получила широкое распространение за счет своей универсальности, малого энергопотребления, возможности реконфигурации и т.д. Сегодня системы на кристалле постепенно заменяют громоздкие вычислительные структуры, реализованные при помощи набора интегральных схем, микроконтроллеры и даже специализированные интегральные схемы.

Современные технологии проектирования SoC имеют два основных направления: платформенный принцип и блочно-модульное построение на основе готовых блоков интеллектуальной собственности так называемых IP-ядер [1].

Блочно-модульный подход позволяет значительно упростить процесс разработки и существенно сократить время проектирования, поскольку IP-ядра представляют собой отлаженные и сертифицированные цифровые блоки, представленные на языке описания аппаратуры (HDL – hardware abstraction level) или в виде готовых файлов для прошивки. Они не содержат ошибок и не требуют тщательной внутренней отладки и верификации. Функциональное назначение таких блоков может быть различным, например, контроллеры интерфейсов, алгоритмы обработки данных, а также процессорные ядра и специализированные модули.

Платформенный подход к проектированию предполагает использование единого программно-аппаратного комплекса, который обеспечивает работу с некоторым перечнем микросхем. Предусматривается применение методов систематического повторного использования стандартных IP-ядер сертифицированных для данной платформы производителем, что избавляет от повторной верификации, при этом возможен импорт ядер сторонних производителей.

Существует два основных направления в реализации SoC: первое – в виде заказных или полузаказных ASIC (Application-Specific Integrated Circuit), изготавливаемых на основе фотошаблонов [3], второе – в базе FPGA (Field Programming Gate Array), которые имеют возможность реконфигурирования за счет смены содержимого конфигурационной памяти.

Основным достоинством ASIC следует считать быстрое действие, малую подверженность внешним воздействиям, в том числе радиации, и низкую стоимость при выпуске крупных партий. Тем не менее, современные технологии изготовления базовых кристаллов постепенно выдвигают FPGA на первый план, поскольку при реализации сложных проектов стоимость ASIC значительно превышает стоимость разработки на FPGA, в то время как скорость обработки на FPGA приближается к скорости обработки ASIC. При этом FPGA позволяет быстро изменять структуру проекта в соответствии с требованиями, полагаясь исключительно на специфику поставленной задачи. Такой подход дает сравнимую с ASIC производительность.

Классический маршрут проектирования SoC подразумевает постепенное нисходящее проектирование от разработки общих алгоритмов системного функционирования, до физического исполнения (рис. 1). Общие алгоритмы разрабатывают и отлаживают с помощью языков высокого уровня, обычно C/C++, либо в среде MATLAB. Переход от алгоритмического описания к описанию на языке представления аппаратуры (HDL) происходит только после верификации на программном уровне.

Данный подход имеет ряд недостатков, связанных с необходимостью преобразовывать алгоритмическое описание в описание на уровне регистровых передач (RTL) вручную. Проблемы возникают из-за различий в представлении системных алгоритмов и описаний на аппаратном уровне, сложности создания и изменения HDL кода, долгой верификации HDL кода, различия между RTL-описанием ПЛИС и ASIC микросхем. Для внесения изменений на системном уровне приходится возвращаться на стадию алгоритмического описания, после чего весь проект требует повторной (или частичной) крупномасштабной отладки и верификации.

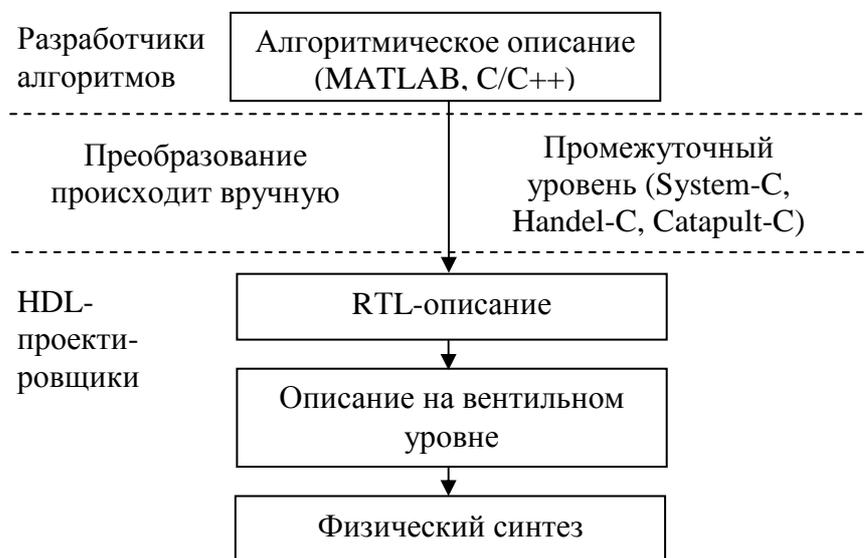


Рисунок 1. Традиционный маршрут проектирования

Сегодня для преодоления разрыва между алгоритмическим и RTL-описанием разработчики активно используют промежуточные C-ориентированные языки описания аппаратуры, такие как System-C, Handel-C или Catapult-C. При этом алгоритмическое описание с языка высокого уровня преобразуется в промежуточный язык, а с промежуточного языка в RTL-описание автоматически или, чаще, полуавтоматически. При этом промежуточный язык проектирования является концептуально близким как к первоначальному алгоритмическому описанию, так и к RTL-описанию. Основным преимуществом такого подхода является высокая скорость трансляции и моделирования (System-C), а также возможность получения вентильного описания из промежуточного при помощи специализированного пакета программ, что позволяет исключить из маршрута RTL-описание системы. Также компактное C-подобное описание позволяет произвести тестирование нескольких вариантов проекта с небольшими материальными затратами. При этом более вероятно найти оптимальный вариант реализации системы. Автоматическая и полуавтоматическая генерация RTL-описания позволяет быстрее вносить изменения в систему, поскольку для этого

требуется только изменить промежуточное представление проекта, либо в алгоритмическое с последующим преобразованием в промежуточное.

Еще одним перспективным направлением является использование ПЛИС-макетов при разработке SoC. Использование аппаратных прототипов будущей системы позволяет осуществить более быстрый и прозрачный переход к HDL-описанию. Таким образом, все ошибки, возникающие на этом этапе, могут быть отслежены до выпуска первой партии микросхем непосредственно с помощью аппаратной модели. Создание макета позволяет быстро перейти к созданию и отладке программного обеспечения системы: драйверов и программных приложений для работы с системой, что сложно сделать при наличии лишь программной модели проектируемого устройства.

Узким местом при использовании прототипов является переход от макета к конкретной реализации, поскольку на практике ПЛИС микросхема во многом отличается от будущей ASIC микросхемы на уровне описания модели на HDL языке, не говоря уже о физических различиях этих структур. Переход к ASIC реализации возможен при одновременной и согласованной разработке ПЛИС и ASIC HDL-моделей с учетом специфики изготовления микросхем каждой из них.

Одной из передовых компаний в области программного обеспечения для такой согласованной разработки полупроводниковых устройств и разработки IP-ядер является Synopsys, Inc. От этой компании, на базе ПЛИС микросхем Xilinx предлагаются специальные платы для прототипирования HAPS (например, HAPS-64 – система прототипирования из серии HAPS-60 построенная на базе 4 микросхем Xilinx Virtex-6 LX760). Также предлагаются различные программные продукты для удобной работы с данными платами, такие как Certify для разработки и декомпозиции ASIC проектов с ПЛИС прототипированием, [Synplify Premier](#) для разработки и отладки ПЛИС систем, [Identify RTL debugger](#) для отладки на уровне регистровых передач.

Методология использования прототипов и макетов, а также методы верификации, разработанные данной компанией вместе с Xilinx позволяют значительно снизить время разработки на основе принципа: проектирование за счет макетирования (Design-for-prototyping). Продуктивность методологии достигается за счет тесного взаимодействия разработчиков прототипов и реализующих структуру SoC-проектировщиков в соответствии с техническими требованиями проекта.

Таким образом, благодаря сходству маршрутов проектирования систем на базовых матричных кристаллах (БМК), ПЛИС и ASIC, широкой доступности целевых кристаллов и средств отладки, постепенно стирается грань между различными подходами в проектировании [2]. Сегодня практика использования промежуточных языков и ПЛИС-прототипирования позволяет снизить время разработки и сократить риск отказа первой производственной партии микросхем. Для проектов реализуемых на FPGA есть возможность подобрать необходимые элементы в виде IP-ядер и выполнить отладку на аппаратном уровне.

Библиографический список

- 1 Бухтеев А.В. Методы и средства проектирования систем на кристалле. // Chip news, 2003, №4, с. 4–14
- 2 Шагурин И. Системы на кристалле. Особенности реализации и перспективы применения // Электронные компоненты. 2009. - № 1. - С. 37-39
- 3 Непомнящий О.В. Сверхбольшие интегральные схемы. Проблемы проектирования / О.В. Непомнящий, Г.А. Скотников, В.А. Хабаров, М.В. Мадудов // Вопросы современной науки и практики. Университет им. В.И. Вернадского. – Тамбов: [б.и.], 2009. – №6(20). – С. 166-173.