

Методы и средства проектирования систем на кристалле

ПРОЕКТИРОВАНИЕ СИСТЕМ НА КРИСТАЛЛЕ ЯВЛЯЕТСЯ УНИВЕРСАЛЬНОЙ И МНОГОПЛАНОВОЙ ДИСЦИПЛИНОЙ, ОБЪЕДИНЯЮЩЕЙ В СЕБЕ МЕТОДЫ ПРОЕКТИРОВАНИЯ ЗАКОНЧЕННЫХ АППАРАТНО-ПРОГРАММНЫХ КОМПЛЕКСОВ, ВСТРАИВАЕМЫХ СИСТЕМ НА ОСНОВЕ СТАНДАРТНЫХ ПРОЦЕССОРОВ И ПРОЦЕССОРНЫХ ЯДЕР, РАЗРАБОТКИ ВСТРОЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРОГРАММИРУЕМЫХ (ПЛИС), ПОЛУЗАКАЗНЫХ И ЗАКАЗНЫХ ИНТЕГРАЛЬНЫХ СХЕМ.

В ОБЩЕМ СЛУЧАЕ, СИСТЕМА НА КРИСТАЛЛЕ МОЖЕТ ВКЛЮЧАТЬ В СЕБЯ РАЗЛИЧНЫЕ ТИПЫ БЛОКОВ: ПРОГРАММИРУЕМЫЕ ПРОЦЕССОРНЫЕ ЯДРА, БЛОКИ ASIC, БЛОКИ ПРОГРАММИРУЕМОЙ ЛОГИКИ, ПАМЯТИ, ПЕРИФЕРИЙНЫХ УСТРОЙСТВ, АНАЛОГОВЫЕ КОМПОНЕНТЫ И РАЗЛИЧНЫЕ ИНТЕРФЕЙСНЫЕ СХЕМЫ. НЕ ОБЯЗАТЕЛЬНО ВСЕ ТАКИЕ БЛОКИ ДОЛЖНЫ БЫТЬ ФИЗИЧЕСКИ РЕАЛИЗОВАНЫ НА ОДНОМ КРИСТАЛЛЕ: ПРОЦЕССОРЫ, БЛОКИ ПАМЯТИ, ПЛИС ИЛИ ASIC МОГУТ ИСПОЛЬЗОВАТЬСЯ КАК ОТДЕЛЬНЫЕ КОМПОНЕНТЫ. В РАМКАХ ИЗЛАГАЕМОЙ МЕТОДОЛОГИИ СТАВИТСЯ ЗАДАЧА СПЕЦИФИКАЦИИ, ВЕРИФИКАЦИИ И ОПТИМИЗАЦИИ СИСТЕМЫ В ЦЕЛОМ, А РЕАЛИЗАЦИЯ ОТДЕЛЬНЫХ УСТРОЙСТВ В КОНТЕКСТЕ ПРОЕКТИРОВАНИЯ РАССМАТРИВАЕТСЯ КАК ПРОЕКТИРОВАНИЕ ЕЁ СОСТАВНЫХ ЧАСТЕЙ.

Данный подход наряду с системами применим и к проектированию отдельных ПЛИС, в которых интегрируется всё большая и большая функциональность, включая процессоры, память, блоки цифровой обработки сигналов, DSP, высокоскоростные входы/выходы и ряд других сложных IP-блоков. Так, разработчики, использующие ПЛИС APEX компании ALTERA, могут выбирать между ядром процессора ARM и интегрированием собственного процессорного ядра Altera NIOS. Таким же образом разработчики, использующие ПЛИС Xilinx Virtex-II Pro, могут интегрировать ядро процессора PowerPC фирмы IBM или собственное процессорное ядро Xilinx MicroBlaze. С другой стороны, производители полужаказных интегральных схем начинают встраивать блоки программируемой логики в ASIC, и разница между этими двумя подходами начинает размываться.

Таким образом, тенденция к сближению архитектур ПЛИС, ASIC и систем с использованием стандартных компонентов в условиях быстро изменяющихся технологических возможностей и экономических факторов, предъявляет требования к унификации маршрутов проектирования. То, что сейчас возможно реализовать только на ASIC, завтра будет можно сделать на ПЛИС, а то, что сегодня делается на базе ПЛИС, надо будет использовать в других проектах, при этом невозможно ответить на вопрос, каков будет способ их реализации. Очевид-

но, что маршрут проектирования должен обеспечивать простой переход от ПЛИС к ASIC с построением прототипов ASIC на основе ПЛИС и наоборот, с переносом центра тяжести в общем объёме проектных работ на системный уровень, абстрагируясь от конкретной реализации.

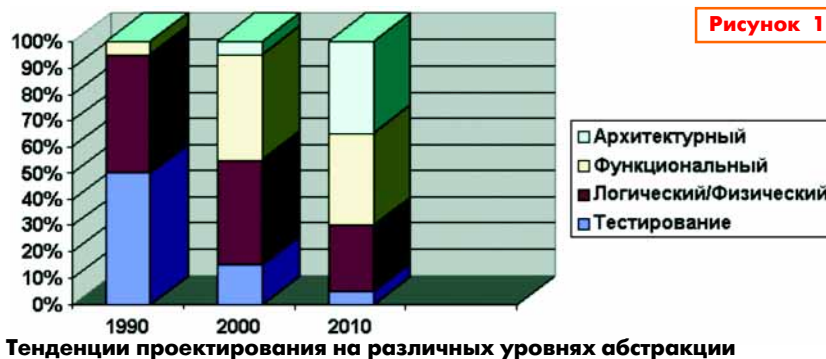
Как следует из анализа состояния и перспектив развития проектирования на различных уровнях абстракции (рис. 1), если в 1990 году реализация проекта (начиная с логического уровня) занимала 90% во всём объёме проектных работ, то в 2000 году эта доля сократилась до 55% и к 2010 году проектирование на архитектурном и функциональном уровнях будет составлять 70% в общем объёме работ, и только 30% придётся на конкретную реализацию проекта в выбранном элементе (библиотечном) базисе.

В любом случае, проектирование систем на кристалле, состоящих из не-

скольких миллионов вентилях, является непростой задачей, и средства проектирования интегральных схем и ПЛИС, непрерывно усложняясь, эволюционируют в сторону системного уровня проектирования. В конечном счёте, чтобы использовать множество IP-блоков и объёмы интегральных схем в несколько миллионов вентилях, нужны соответствующие средства проектирования, позволяющие использовать все эти возможности в разработках.

Общий маршрут проектирования систем на кристалле показан на рис. 2 и состоит из следующих основных этапов:

- концептуальное проектирование системы; основной задачей данного этапа является исследование проектируемой системы и получение её исполняемых спецификаций на языке высокого уровня (стандартно на C/C++);



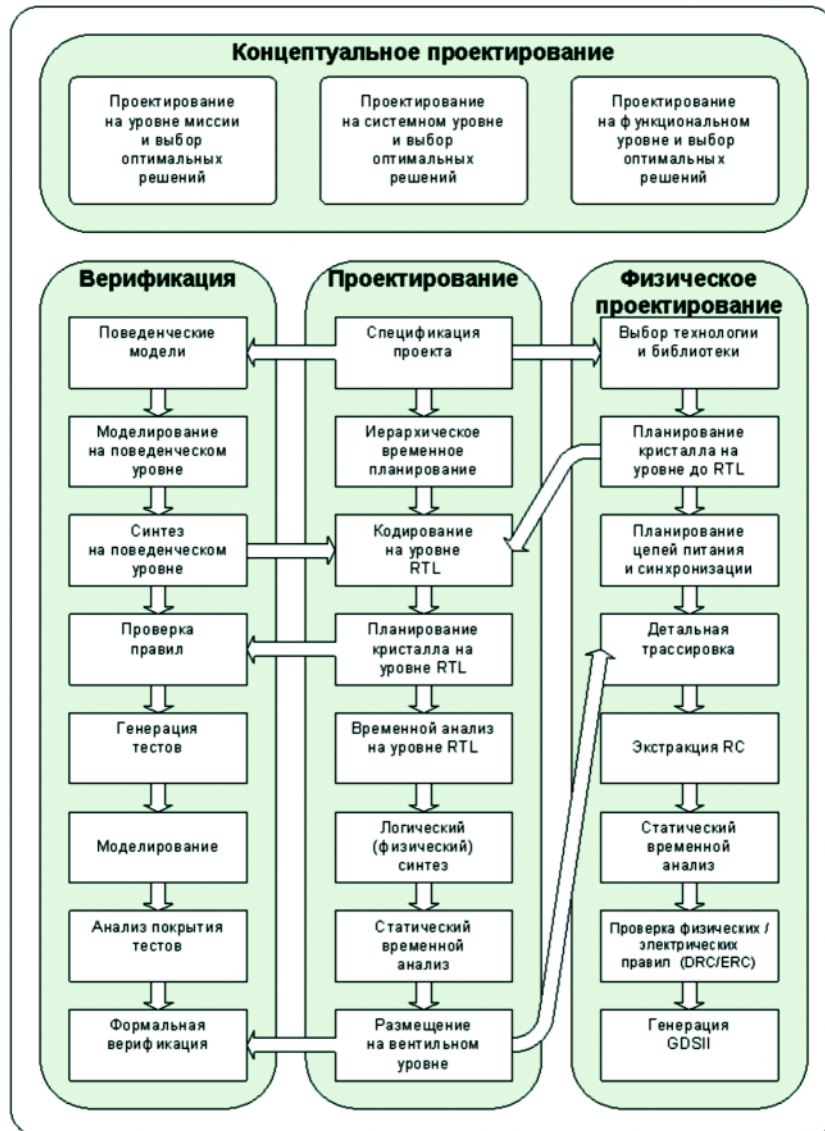


Рисунок 2 Общий маршрут проектирования

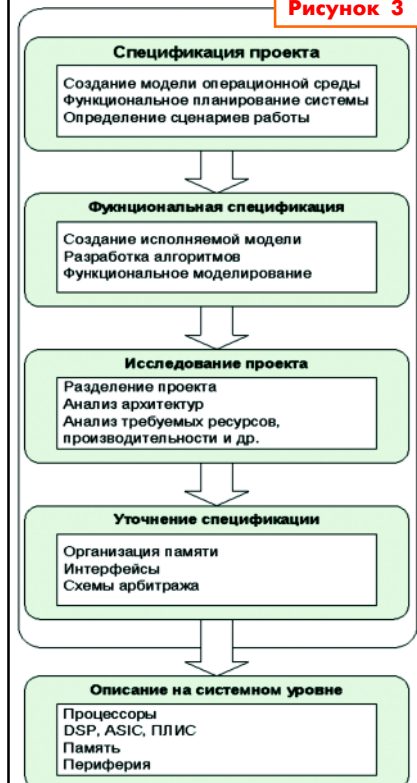
- проектирование, то есть трансформация исполняемой спецификации проекта на уровень регистровых передач (получение спецификаций на языках Verilog/VHDL) и далее на вентиляный уровень;
- верификация проекта, то есть проверка проекта и проектных решений на соответствие исходной спецификации и другим требованиям в процессе проектирования и детализации;
- физическое проектирование, начиная от выбора технологического и библиотечного базиса и заканчивая получением финального описания проекта в формате GDSII.

При проектировании систем на кристалле концептуальный уровень является критическим для оценки общих харак-

теристик системы. На этом уровне создаётся общая исполняемая спецификация проектируемой системы, позволяющая исследовать и оценить различные варианты её построения и выбрать оптимальное решение, которое будет реализовано в дальнейшем. Здесь решаются следующие задачи:

- создание функциональной модели системы, то есть описание системы с точки зрения тех алгоритмов и функций, которые она должна выполнять, без привязки к способам их реализации;
- моделирование системы в её операционной среде (на уровне «миссии») с реальными данными и сигналами (аудио- и видеоинформацией, радиоканалами, расположением и движением объектов и др.)

Рисунок 3



Маршрут проектирования на концептуальном уровне

- определение архитектуры системы с точки зрения необходимых ресурсов и их организации для программно-аппаратной реализации функциональной модели.

Таким образом, имея исполняемую спецификацию системы, поведенческие модели и общую архитектуру, проектирование, верификация и топологическая реализация системы далее ведутся параллельно.

КОНЦЕПТУАЛЬНЫЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ

Основной целью в процессе спецификации проекта является определение и спецификация основных функций системы и создание исполняемой системной модели. С использованием моделирования, эта системная модель используется для верификации корректности работы системы с функциональной точки зрения в операционной среде, в которой она должна работать, а также для определения необходимых аппаратных ресурсов для работы и архитектуры системы. Общий маршрут проектирования на данном этапе приведён на рис. 3.

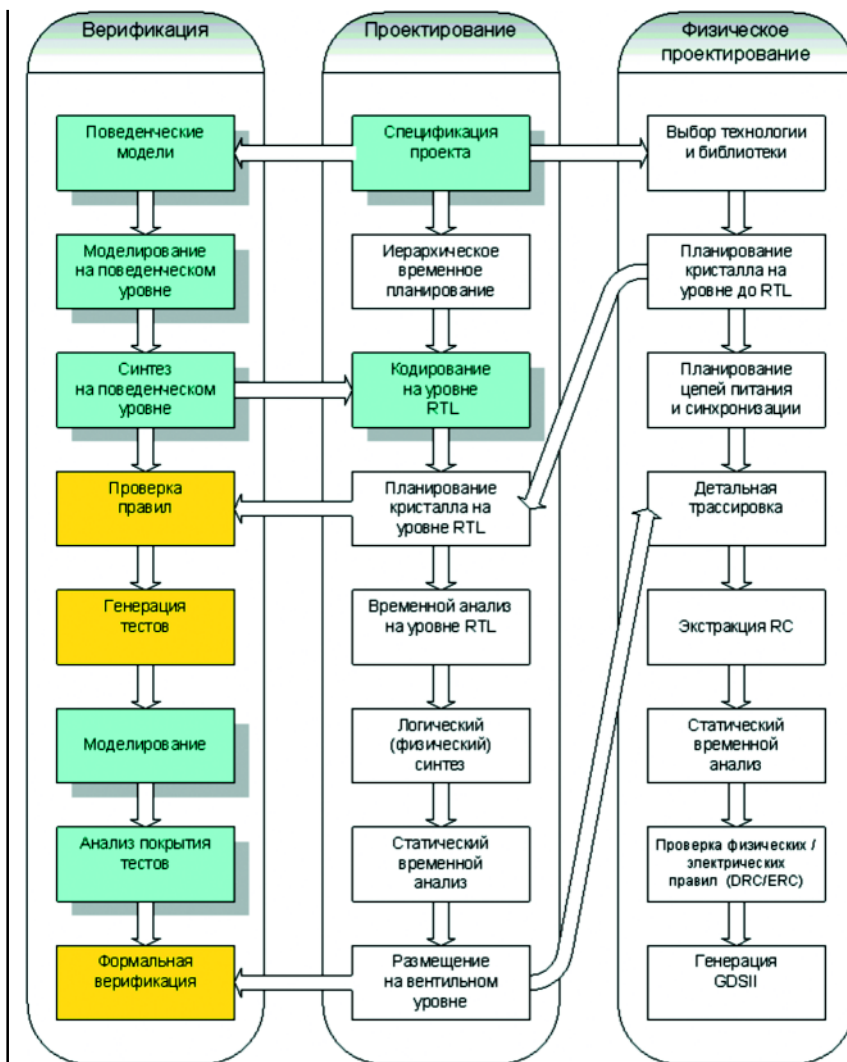


Рисунок 4 Этапы верификации в общем маршруте проектирования

На этапе общей спецификации проекта определяется операционная среда, в которой должна работать система, основные сценарии работы, общие функциональные характеристики и протоколы. Так, здесь могут моделироваться различные электронные приводы, системы управления, зоны покрытия, взаимное расположение и движение объектов, например, для распределённых систем типа «базовые станции – мобильные терминалы» и другое.

Далее создаётся функциональная спецификация системы, целью которой является определение и моделирование функционирования системы с точки зрения выполняемых алгоритмов. Здесь может быть задано и промоделировано поведение всей системы в целом или её отдельных блоков. Как правило, на этом уровне функции системы моделируются с реальными данными и сигналами. Например, можно описать цифро-

вой приёмник-передатчик и протестировать его с использованием реалистичной модели радиотракта.

На этапе исследования проекта моделируемые функции трансформируются и разделяются для выполнения на ряде платформ, или архитектур, которые содержат различные наборы компонентов, такие как программируемые процессоры, память, ASIC, ПЛИС или блоки системы на кристалле. Используя различные виды оценок, целью здесь является нахождение оптимальной архитектуры, которая отвечает заданным критериям, таким как работа в реальном времени, производительность, стоимость, потребляемая мощность. Программные функции оцениваются с точки зрения размера кода и наихудшего времени выполнения, измеряемого количеством тактов процессора, а аппаратные функции — в количестве эквивалентных вентилях.

Наконец, производится уточнение спецификации системы, где создаётся более детальное описание системной архитектуры, которая передаётся на проектирование. Такое описание на системном уровне может содержать некоторые детали последующей реализации, но функциональная часть состоит из поведенческих моделей на языках C/C++/SystemC. Далее уже используется совместное программно-аппаратное проектирование с применением моделей конкретных процессоров и шин (функциональных моделей), блоков, описанных на языках проектирования аппаратуры VHDL/Verilog и так далее.

Система проектирования MLDesigner компании MLDesign Technologies (www.mldesigner.com) позволяет решать все эти задачи, а именно:

- проектирование на уровне миссии и выбор оптимальных решений: моделирование «операционной» среды, определение статических и динамических сценариев, планирование целевых задач и моделирование среды их выполнения (например, топография местности, модели каналов), тестирование сложных проектов на уровне целевых задач;
- проектирование на системном уровне и выбор оптимальных решений: моделирование и анализ производительности, цифровая обработка сигналов и архитектура DSP, сетевая архитектура и протоколы, компьютерные архитектуры, встроенные системы;
- проектирование на функциональном уровне и выбор оптимальных решений: выбор оптимальных алгоритмов, выбор оптимальных протоколов, анализ алгоритмов с фиксированной точкой, программное/аппаратное разделение, выбор оптимальной реализации, схемы разрешения конфликтов на шинах, методы управления памятью.

ПРОЕКТИРОВАНИЕ И ФУНКЦИОНАЛЬНАЯ ВЕРИФИКАЦИЯ

Функциональная верификация занимает всё более важное место в общем маршруте проектирования. Если раньше под проектированием понималась разработка проекта на уровне регистровых передач (и далее переход на вентилярный уровень средствами логического синтеза), а верификация проводилась средствами логического моделиро-

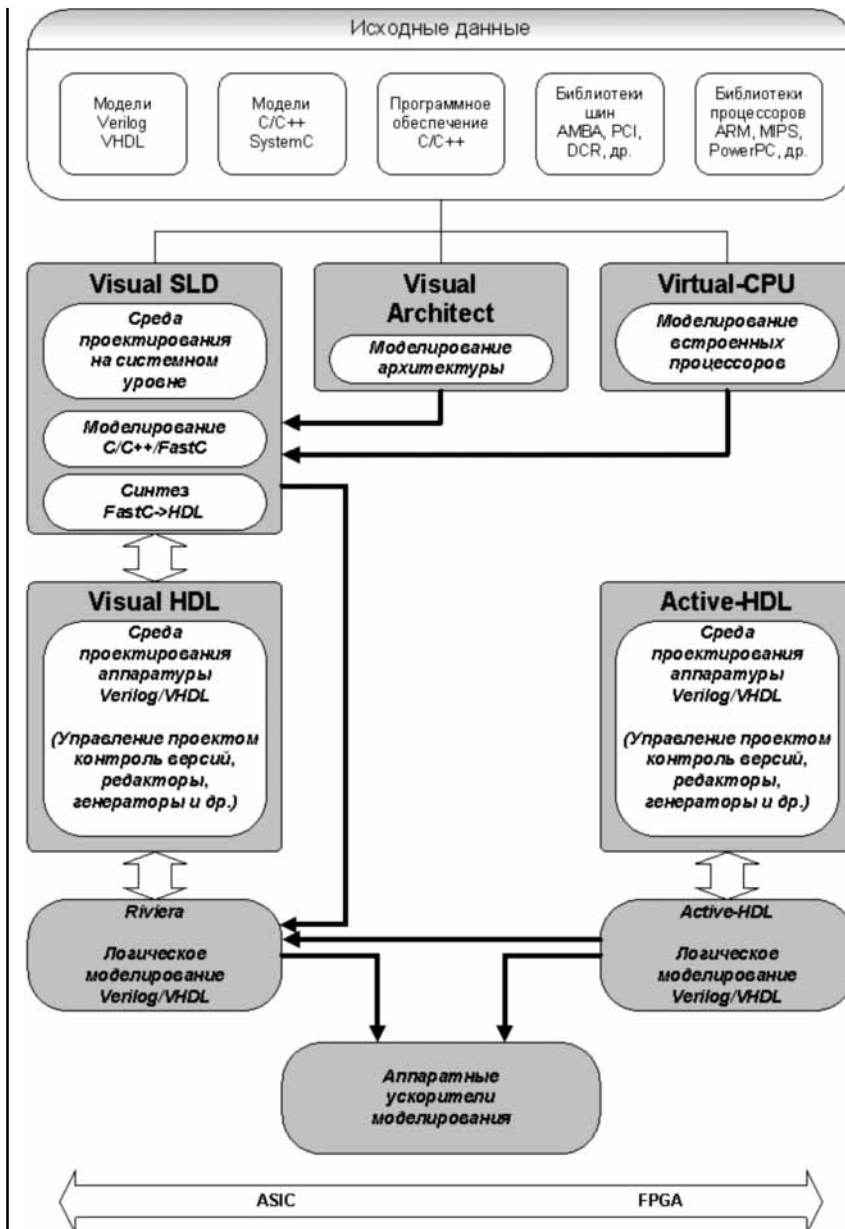


Рисунок 5 Маршрут проектирования в системе VisualElite

вания, то сейчас верификация начинается на поведенческом уровне на стадии разработки общей спецификации проекта. Место данного этапа в общем маршруте проектирования показано на рис. 4.

Основными требованиями, предъявляемыми к составу средств функционального проектирования и верификации, являются:

- анализ архитектуры, производительности и других системных параметров проектируемых систем;
- проектирование аппаратно-программных систем, то есть возможность совместной разработки и верификации аппаратуры и встроенного программного обеспечения;
- проектирование систем с использованием процессорных блоков, то есть использование моделей процессоров при разработке аппаратуры и программного обеспечения;
- единая среда проектирования, от системного уровня до уровня регистровых передач и вентильного уровня с поддержкой языков C, C++, SystemC уровней 1.0 и 2.0 и языков описания аппаратуры Verilog и VHDL;
- наличие библиотек и высокоуровневых конструкций для функциональных блоков и коммуникационных каналов, включая таблицы связности;
- средства управления данными и документирования проектов.

Методология проектирования в системе проектирования VisualElite компании SUMMIT DESIGN (www.sd.com) предоставляет системным архитекторам, разработчикам и программистам единую среду проектирования для спецификации, верификации и анализа архитектуры и функционирования от системного уровня до уровня регистровых передач (рис. 5).

Типичная система на кристалле состоит из интерфейса внешней шины, возможно, встроенного процессора, памяти "на кристалле" (или широкополосного интерфейса к внешней памяти), ряда функциональных модулей и шины "на кристалле" (*On-chip Bus*, OCB), которая их соединяет. И первая решаемая задача — это анализ архитектуры системы и её производительности. По мере получения исполняемых спецификаций аппаратуры и программного обеспечения, проверить их взаимодействие и архитектуру системы можно в Virtual-CPU, которая может быть сконфигурирована для любого процессора или шины.

После того, как необходимые итерации спецификации архитектуры проделаны и отвергнуты, реализация аппаратуры и программного обеспечения может продолжаться параллельно, и на этом этапе необходимо проектирование аппаратно-программных интерфейсов для создания и управления системными регистрами, которые устанавливают аппаратно-программный интерфейс и логически разделены между периферийными аппаратными устройствами в системе и прикладным программным обеспечением, исполняемым во встроенных ядрах.

На высших уровнях представления используются языки C/C++ или SystemC. Для моделирования кода C/C++ используется встроенное ядро моделирования, которое осуществляет планирование и исполнение моделирования в соответствии со структурой и поведенческими функциями проекта вместе с программными объектами, например, такими как операционные системы. Когда модули C/C++/SystemC постепенно детализируются с переходом на нижние уровни абстракции, разработчик может заместить модуль C представлением HDL (Verilog и VHDL) или синтезировать его, если данный блок представлен на уровне C в соответствии с определёнными правилами.

Редактор блок-диаграмм является основным редактором, который поддерживает блоки, написанные на C/C++,

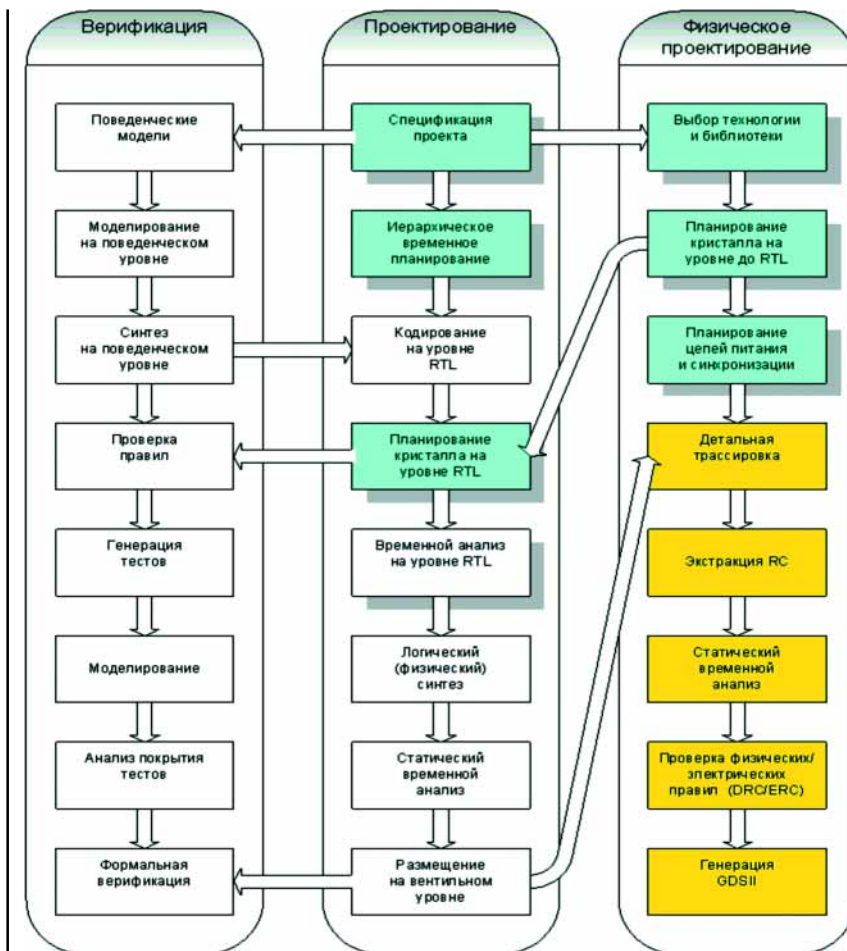


Рисунок 6 Этапы архитектурного планирования в общем маршруте проектирования

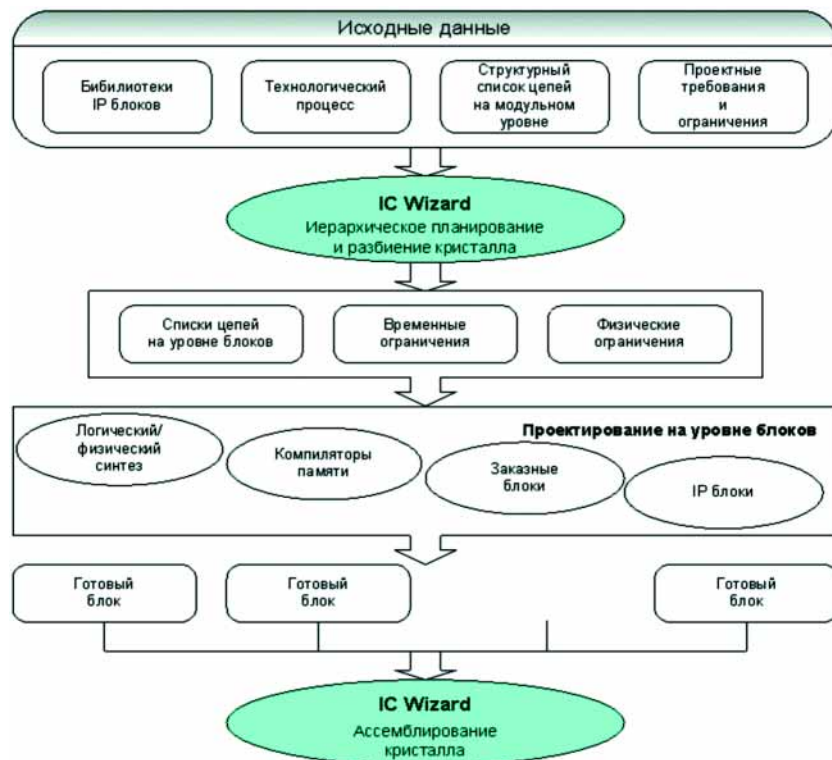


Рисунок 7 Архитектурное планирование кристалла в системе IC Wizard

SystemC и блоки HDL. Он предоставляет возможности построения иерархии и параллельных процессов, вместе с другими аппаратными особенностями RTL, такими как узлы, пины, компоненты и блоки. Блоки могут описывать алгоритмы на уровне транзакций, тактовую семантику или некоторое детальное событийное поведение. Для определения семантики таких параллельных блоков предоставляется компактный набор конструкций типа операторов чувствительности, временных функций и набор предопределённых функций чтения/записи, для различных структур данных и протоколов.

Между блоками поток данных контролируется посредством коммуникационных каналов. Каждый канал реализует определённый протокол, такой как очередь, защищённый канал связи или аппаратный сигнал, поддерживаемые библиотеками классов C++. Однако, пользователи могут определять и добавлять свои собственные коммуникационные каналы, используя специальный пакетный механизм, который обеспечивают как C, так и C++ пакеты.

Когда модули C/C++/SystemC постепенно детализируются с переходом на нижние уровни абстракции, Visual Elite без дополнительной настройки ассоциирует коммуникационные каналы блока C с сигналами HDL, обеспечивая, таким образом, единую среду проектирования и верификации от системного уровня до уровня реализации.

АРХИТЕКТУРНОЕ ПЛАНИРОВАНИЕ КРИСТАЛЛА

Архитектурное планирование требуется для всех типов кристаллов, как для систем на кристалле, которые строятся из IP-блоков, так и больших (несколько миллионов вентилей) ASIC. Место данного этапа в общем маршруте проектирования показано на рис. 6.

Маршрут проектирования систем на кристалле используется, когда применяются IP-блоки или выполнение проекта распределяется между несколькими группами (компаниями). В данном маршруте логический проект уже имеет заданное физическое разделение, когда блоки остаются едиными как логически, так и физически. Типичный проект системы на кристалле состоит из ряда блоков верхнего уровня, которые физически размещаются на этапе архитектурного планирования кристалла, а за-

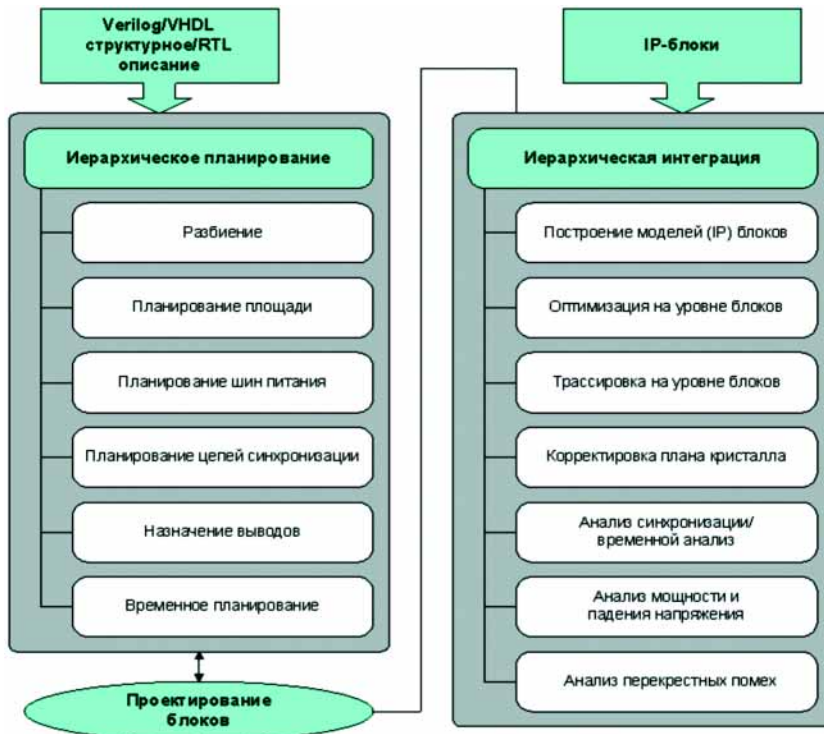


Рисунок 8 Иерархическое планирование и интеграция в IC Wizard

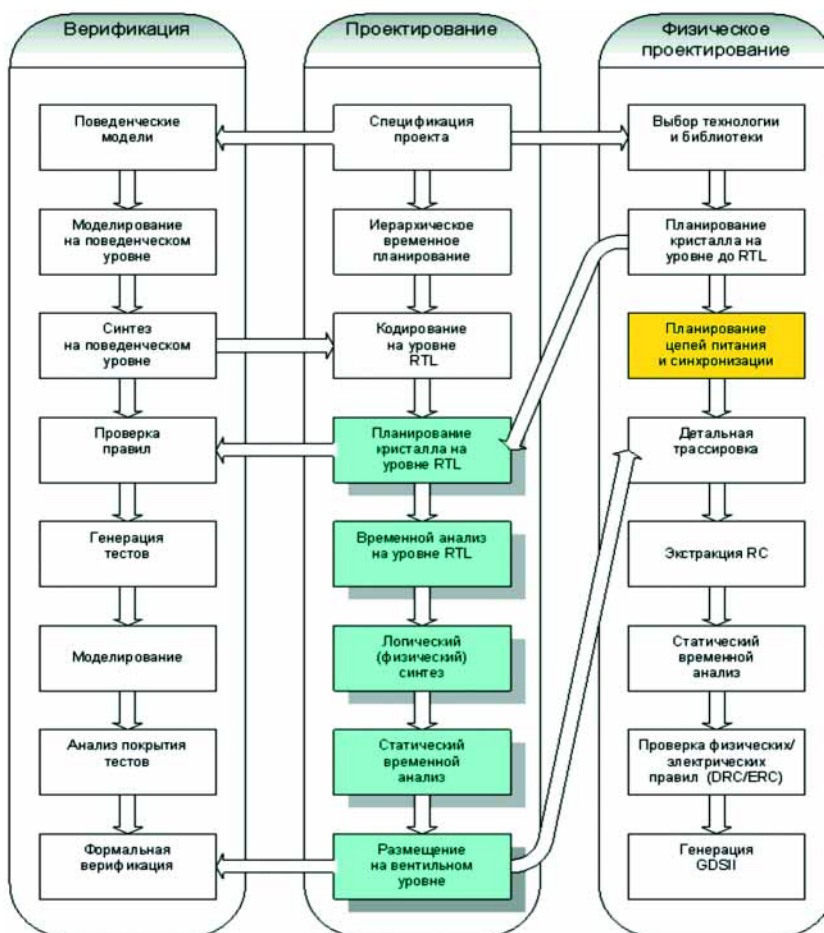


Рисунок 9 Этапы создания физического прототипа в общем маршруте проектирования

тем проект трассируется с использованием каналов между блоками.

При проектировании ASIC, когда нет заданной физической иерархии, архитектурное планирование применяется для проектов, слишком больших для типичных процедур проектирования (синтеза, размещения и трассировки). Целью архитектурного планирования в данном маршруте является разбиение кристалла на блоки, которые затем могут быть спроектированы независимо друг от друга, как если бы они являлись отдельными кристаллами. Для каждого такого блока генерируется набор граничных требований, включая его конфигурацию, расположение портов и временные ограничения. Затем производится ассемблирование проекта, аналогично системе на кристалле.

Пакет IC Wizard компании MONTEREY DESIGN SYSTEMS (www.montereydesign.com), маршрут проектирования в котором показан на рис. 7, поддерживает архитектурное планирование для всех типов проектов и позволяет оценить на ранних стадиях проектирования, ещё до того, как будет готово описание всего проекта на языках VHDL/Verilog на уровне регистровых передач, такие характеристики физического проекта, как производительность, площадь, соотношение сторон, схемы питания и синхронизации, падение напряжения и эффекты электромиграции.

Задача архитектурного планирования состоит из двух составных частей: с одной стороны, это собственно планирование, связанное с разбиением проекта на блоки, с другой стороны, это иерархическая интеграция проекта на основе блоков. Задачи, решаемые в системе IC Wizard на этапах иерархического планирования и интеграции, показаны на рис. 8.

Таким образом, IC Wizard может использоваться на всех стадиях выполнения проекта, от определения архитектуры кристалла до конечной физической реализации:

- для создания архитектурного плана кристалла на начальных стадиях выполнения проекта (до описания проекта на уровне регистровых передач) и оценки его ключевых характеристик;
- для многоуровневого иерархического разбиения проекта на блоки с генерацией списков цепей, физических и временных ограничений для каждого блока;
- по мере готовности физического прототипа или топологии для отдель-

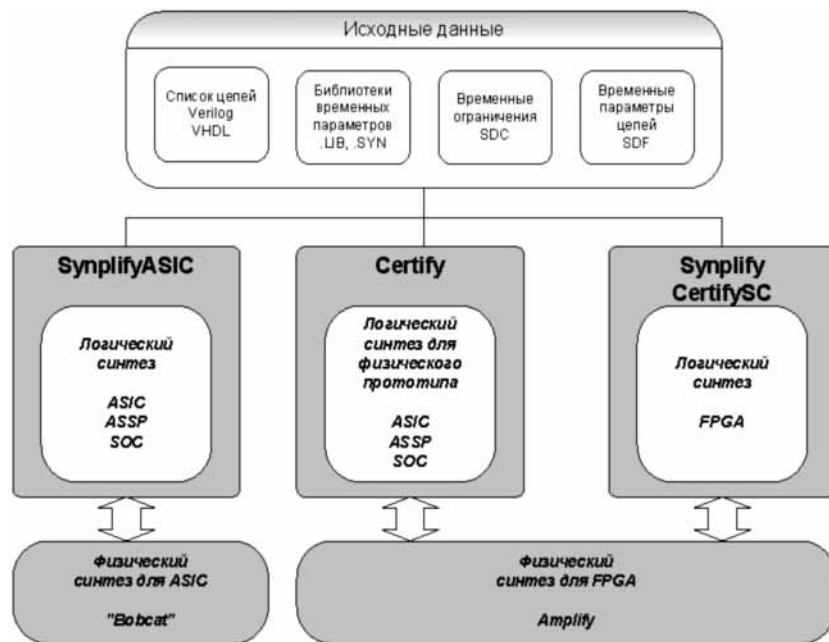


Рисунок 10 Логический синтез для различного типа реализаций

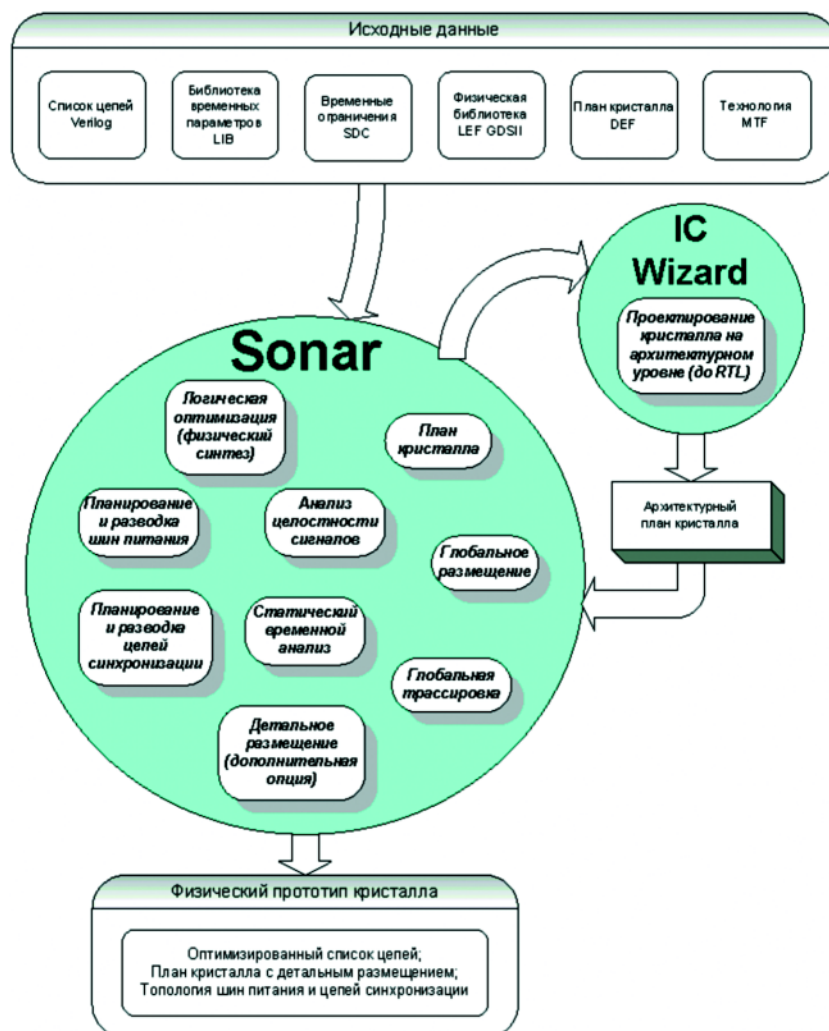


Рисунок 11 Проектирование физического прототипа в системе Sonar

ных блоков для модификации архитектурного плана кристалла и уточнения физических и временных требований к другим блокам;

- для трассировки всех глобальных (межблочных) соединений, включая цепи питания и синхронизации, и асемблирования кристалла.

Архитектурное планирование также позволяет разработчикам систем на кристаллах сделать оценки, связанные с использованием существующих IP-блоков, заданных как на уровне регистровых передач, так и на уровне готовой топологии. Выбор окончательной конфигурации может быть сделан за счёт быстрой подстановки различных IP-блоков и оценки их влияния на общие системные параметры.

Архитектурное планирование является первым шагом на пути к предсказуемому успешному созданию систем на кристаллах. Главной его задачей является быстрое определение физического разбиения кристалла, требований к отдельным блокам и управление взаимодействием между всеми подсистемами в процессе проектирования, начиная с определения архитектуры и до получения финальной топологии.

ЛОГИЧЕСКИЙ СИНТЕЗ И ПРОЕКТИРОВАНИЕ ФИЗИЧЕСКОГО ПРОТОТИПА

Целью данного этапа проектирования является создание из исходного описания всего проекта или его отдельных блоков на уровне регистровых передач на языках Verilog и/или VHDL списка цепей в базе библиотечных элементов производителя и физического прототипа проекта/блоков. Место данного этапа в общем маршруте проектирования показано на рис. 9.

Описание проекта на уровне регистровых передач на языках Verilog или VHDL в целом технологически независимо, хотя до разработки RTL-кода необходимо принимать во внимание последующую реализацию, если речь идёт о проектировании системы на кристалле. Поэтому логический синтез является ключевым и универсальным инструментом при проектировании цифровых систем и их реализации в виде ПЛИС, физических прототипов как на основе ПЛИС (макетов), так и виртуальных прототипов кристаллов. Здесь прототипы играют разную роль — в то время, как макеты на основе ПЛИС используются для функци-

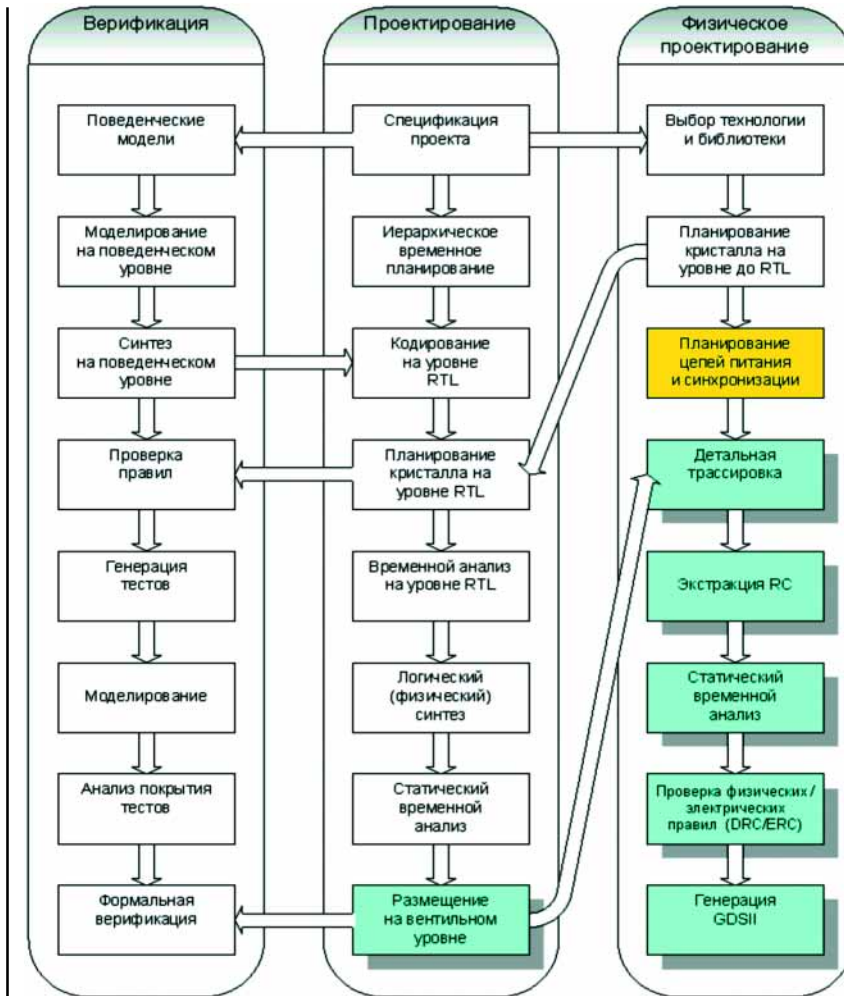


Рисунок 12 Этапы разработки физического прототипа в общем маршруте проектирования

ональной верификации, виртуальный физический прототип используется для определения всех параметров топологической реализации кристалла.

Логический синтез

Наиболее полное и унифицированное решение в области логического синтеза принадлежит фирме SYNPLICITY (www.synplicity.com), которое покрывает все аспекты проектирования, связанные с логическим синтезом (рис. 10). Таким образом, достигается технологическая независимость — проектирование ПЛИС, ASIC и макетирование ASIC на основе ПЛИС без внесения каких-либо изменений в исходные спецификации проектов на уровне RTL (VHDL и/или Verilog) и командные файлы с “автоматическим” переходом с одной технологии на другую.

Независимость от конкретного производителя ПЛИС и возможность повторного использования спроектированных модулей в любом элементном ба-

зисе является ключевым моментом при разработке IP-блоков.

При проектировании систем на кристаллах, ASIC или отдельных блоков (IP) для их финальной верификации в маршруте проектирования применяется их макетирование (эмуляция). Это связано с невозможностью промоделировать проекты такого объема за приемлемое время и, особенно для проектов в области телекоммуникаций, мультимедиа и др., необходимостью верификации проектов в реальной среде с огромными потоками обрабатываемой информации. При этом для макетирования одной ASIC может потребоваться от одной до нескольких десятков ПЛИС, в зависимости от сложности проекта. Кроме того, для прототипирования ASIC существует ряд готовых печатных плат, которые позволяют не только построить макет ASIC, но и провести комплексную отладку и верификацию макета в составе вычислительного комплекса с использованием традиционных систем логического моделирования. Основное отличие системы

синтеза для физического прототипа от системы синтеза ПЛИС заключается в возможности разбиения проекта на необходимое для реализации проекта количество ПЛИС без внесения изменений в исходное описание проекта и сохранения целостности его функциональных и временных характеристик.

Дополнительные модули физического синтеза позволяют учесть физические особенности реализации, а именно физические ограничения по проекту, наряду с временными ограничениями, и синтезируют план кристалла, содержащий физическую информацию. Например, при синтезе ASIC, используемые сейчас модели нагрузок (WLM) на технологиях ниже 0,18 микрон дают неудовлетворительные результаты. Применение физического синтеза позволяет синтезировать логическую схему, соответствующую физической реализации, используя вместо WLM модели нагрузок, рассчитанные исходя из физической информации о размещении элементов.

Проектирование физического виртуального прототипа

Физический виртуальный прототип является представлением проекта системы на кристалле, ASIC или отдельного блока, которое доступно до финальной топологии и содержит достаточную физическую информацию, чтобы точно предопределить все параметры, такие как временные характеристики, занимаемую площадь, потребляемую мощность и другие.

Физический прототип должен быть достаточно точным, чтобы он мог быть передан на проектирование топологии в полной уверенности, что конечная реализация будет отвечать всем проектным требованиям.

На рис. 11 показан маршрут проектирования в системе Sonar компании MONTEREY DESIGN SYSTEMS (www.montereydesign.com). В данном маршруте отсутствует модульность и последовательность выполняемых проектных задач, поскольку здесь проводится глобальная и параллельная оптимизация проекта по всем аспектам проектирования, включая физические эффекты проектирования на глубоких субмикронных и нанотехнологиях. Таким образом, задача построения физического прототипа, который может быть использован для проектирования всего кристалла или его отдельных блоков, является комбинацией задач оптимизации (включая логическую оптимизацию), глобальной топологической реализации (глобального размещения и

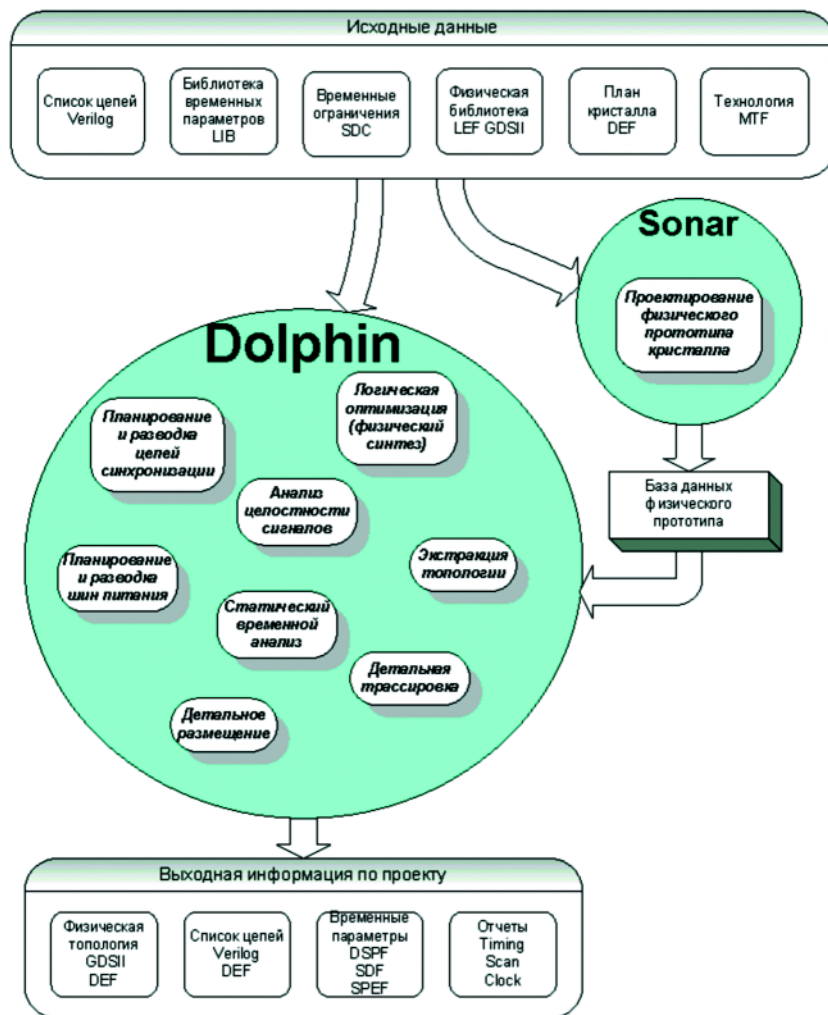


Рисунок 13 Маршрут физического проектирования в системе Dolphin

трассировки, разводки шин питания и синтеза цепей синхронизации) и анализа. Далее физический прототип передается на физическую реализацию, где выполняются наиболее длительные по времени выполнения процедуры синтеза физической топологии и её верификации.

Основной задачей проектирования физического прототипа является решение всех проблем, которые могут возникнуть при проектировании топологии перед тем, как оно будет начато, и передача его на финальную реализацию.

Проектирование физического прототипа может использоваться совместно с архитектурным планированием для проверки проектных требований и ограничений. При этом будут определены любые проектные требования, которые невозможно реализовать, и информация будет передана назад для внесения изменений в план кристалла. Затем будет сделана логическая оптимизация, глобальное размещение и трассировка, разводка шин питания и цепей синхронизации и анализ проекта.

Таким образом, результатом будет являться гораздо более точное представление блоков, чем было доступно ранее. Эта информация передается назад планировщику, так что архитектурный план кристалла будет инкрементно обновляться, и соответственно уточняться требования к другим блокам.

ПРОЕКТИРОВАНИЕ ФИЗИЧЕСКОЙ ТОПОЛОГИИ ПОЛУЗАКАЗНЫХ СХЕМ

Этап проектирования физической топологии ИС заключается в получении описания топологии в формате GDSII для передачи его на производство из исходного списка цепей. Место данного этапа в общем маршруте проектирования показано на рис. 12.

Традиционный подход к задаче физического проектирования топологии ИС состоит из целого ряда последовательных операций (состав которых определяется используемой технологией, в

зависимости от тех физических эффектов, которые необходимо учитывать при проектировании) со множеством итераций, поскольку по результатам проектирования на каждом этапе необходимо бывает вносить изменения и проводить перепроектирование на предыдущих стадиях, и, кроме того, требует детальных знаний технологии и опыта проектирования.

Система физического проектирования Dolphin компании MONTEREY DESIGN SYSTEMS (www.montereydesign.com) с использованием "технологии глобального проектирования" позволяет сразу спроектировать топологию ИС за один проход, выполняя параллельно задачи размещения, трассировки, оптимизации логики и временных соотношений, разводку цепей питания и синхронизации, экстракцию топологии и анализ сигналов.

На рис. 13 показан маршрут проектирования в системе Dolphin. В данном маршруте отсутствует модульность и последовательность выполняемых проектных задач, поскольку здесь проводится глобальная и параллельная оптимизация проекта по всем аспектам проектирования, включая физические эффекты проектирования на глубоких субмикронных и нанотехнологиях.

Предварительно, до этапа физического проектирования, может использоваться система проектирования физического прототипа кристалла Sonar, база данных которого может быть непосредственно загружена в Dolphin, или использоваться план кристалла в формате DEF, полученный в любом планировщике. Как Sonar, так и Dolphin могут использоваться с системой архитектурного планирования IC Wizard для проектирования, соответственно, физического прототипа и физической топологии отдельных блоков, с разбиением всего проекта на блоки и их последующей интеграцией в IC Wizard.

Наконец, в конфигурации Dolphin RTL, включающей в себя составной частью оптимизированную для системы проектирования физической топологии Dolphin версию системы логического синтеза SynplifyASIC, обеспечивается полный маршрут проектирования, начиная от заданного описания проекта на уровне регистровых передач до готовой топологии в формате GDSII.

За любой дополнительной информацией можно обращаться в компанию STREAMLINE DESIGN SOLUTIONS по адресу sasha@streamline-ds.net или телефону (095) 506-1474 или в офис её российского партнёра — компанию "Электрейд-М" (www.eltm.ru).