

ПЛАНИРОВАНИЕ КРИСТАЛЛА С МНОГОУРОВНЕВОЙ ФИЗИЧЕСКОЙ ИЕРАРХИЕЙ С ПОМОЩЬЮ ИНСТРУМЕНТА IC Compiler II

К.Розе, Д.Радченко

УДК 621.382
БАК 05.27.00

Проектирование сложных систем на кристалле (СНК) требует применения иерархического подхода к созданию топологии кристалла. В большинстве САПР одновременно возможно использовать всего два уровня физической иерархии, что затрудняет реализацию проектов с многоуровневой иерархией, вызывая риски увеличения сроков разработки топологии или снижения качества конечного результата. Инструмент физического проектирования нового поколения IC Compiler II от компании Synopsys позволяет автоматизировать процесс разработки планировки кристалла с многоуровневой физической иерархией за счет применения новой модели данных, поддерживающей многоуровневую физическую иерархию. Мы рассмотрим ключевые вопросы разработки планировки кристалла (Floorplanning) с многоуровневой физической иерархией и обсудим, как с помощью IC Compiler II можно решить эти проблемы и получить оптимальный результат за минимальное время.

Сложность СНК постоянно растет. Системы обработки потокового видео высокого разрешения, видеонаблюдения, навигационные системы и системы реального времени – это лишь несколько примеров современных приложений, требующих высокой производительности. С одной стороны, это вызывает необходимость в интегрировании на кристалле все больших вычислительных ресурсов, но, с другой стороны, развитие технологических процессов полупроводникового производства дает возможность инженерам размещать на кристалле все более широкий функционал (рис.1).

Физическая реализация таких СНК сложна: разработчики имеют дело со схемами, работающими на частотах, близких к гигагерцовым, и включающими сотни миллионов эквивалентных логических вентиля и десятки миллионов размещаемых на кристалле объектов (рис.2).

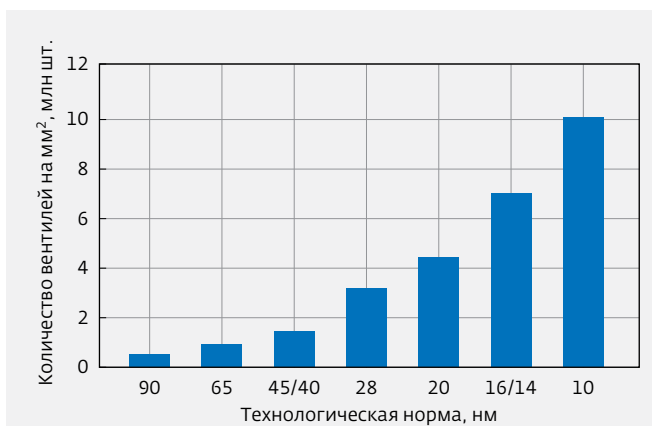


Рис.1. Количество используемых вентиля на мм² при различных топологических нормах. Источник: International Business Strategies, Inc. (IBS), Производственные технологии и экосистемы (Process Technology and Ecosystems). Май 2015 года

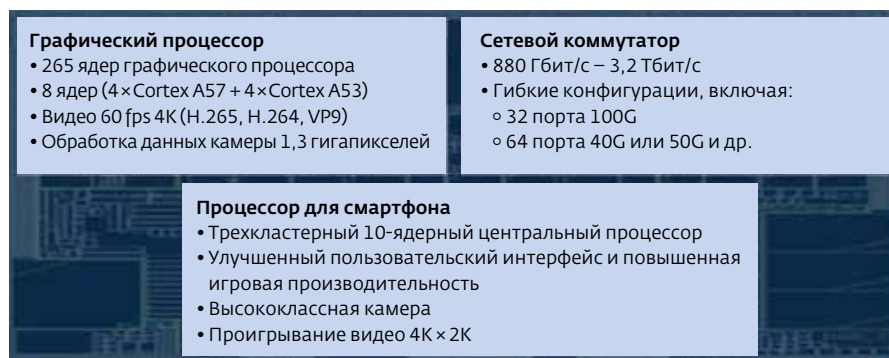


Рис.2. Новейшие СнК, упоминавшиеся в новостях

Для того чтобы успешно справиться с этими задачами, необходим иерархический подход к реализации таких проектов – разделение проекта на блоки. Поскольку эти блоки сами по себе крупные и сложные, требуется вторичное их дробление на более мелкие блоки (рис.3). Обычной практикой в этом случае является использование трех, четырех или даже пяти уровней физической иерархии кристалла. Для каждого из них необходимо определить размеры и геометрические формы, распо-

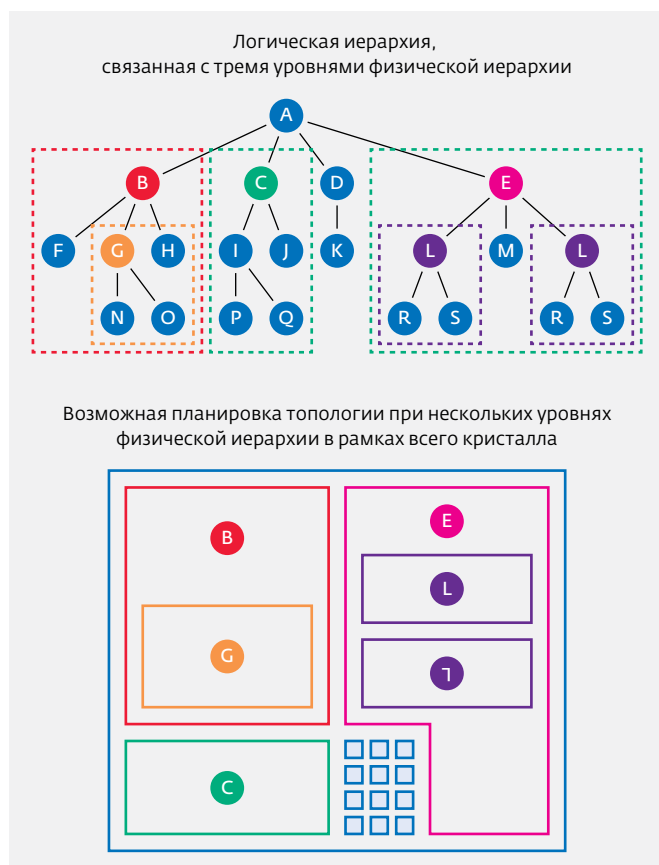


Рис.3. Физическая планировка, разделенная на блоки физической иерархии трех уровней

ложение портов входных и выходных сигналов, топологию синхросигналов, шин питания и земли, определить временные ограничения интерфейсных сигналов.

Дополнительные сложности могут возникать, когда инженеры используют так называемые многократно используемые блоки (Multiply-Instantiated Block – MIB). При реализации MIB следует учитывать индивидуальное (контекст-

ное) окружение каждой копии такого блока, чтобы оптимально определить его форму, размеры, размещение портов входных и выходных сигналов, правильно определить временные ограничения для интерфейсных сигналов, наиболее оптимально удовлетворяющие всем индивидуальным окружениям такого блока. Кроме того, MIB может быть использован на различных уровнях физической иерархии.

Большинство инструментов физического проектирования поддерживает одновременно не более двух уровней физической иерархии: верхний уровень и уровень блоков. Поэтому разработчикам приходится использовать рекурсивный метод для планирования и анализа многоуровневых иерархических проектов.

НЕДОСТАТКИ ДВУХУРОВНЕВОЙ МЕТОДОЛОГИИ ПРОЕКТИРОВАНИЯ

При двухуровневом подходе к планированию кристалла разработчики вынуждены задавать форму блоков, размещать порты и определять требования к временным параметрам блока в каждый момент в рамках лишь одного уровня (рис.4). При подходе к планированию "сверху вниз" планирование нижних уровней в контексте всего кристалла можно выполнить только на основе требований и ограничений, жестко установленных на верхнем уровне. При использовании подхода "снизу вверх" ограничения и проектные решения блочных уровней становятся жесткими, неизменяемыми ограничениями при проектировании верхнего уровня. Разработчики, отвечающие за разные физические уровни проекта, должны согласовывать между собой любые изменения на всех затрагиваемых уровнях и находить решения, направленные на минимизацию необходимой доработки на других уровнях проекта (рис.5). Эти согласования требуют дополнительного времени, что трудно учитывать в графике выполнения работ по проекту. Часто это является основной причиной срыва сроков работ.

Встает вопрос: как повысить эффективность и достичь оптимального качества результатов при физической реализации проекта со сложной иерархической структурой?

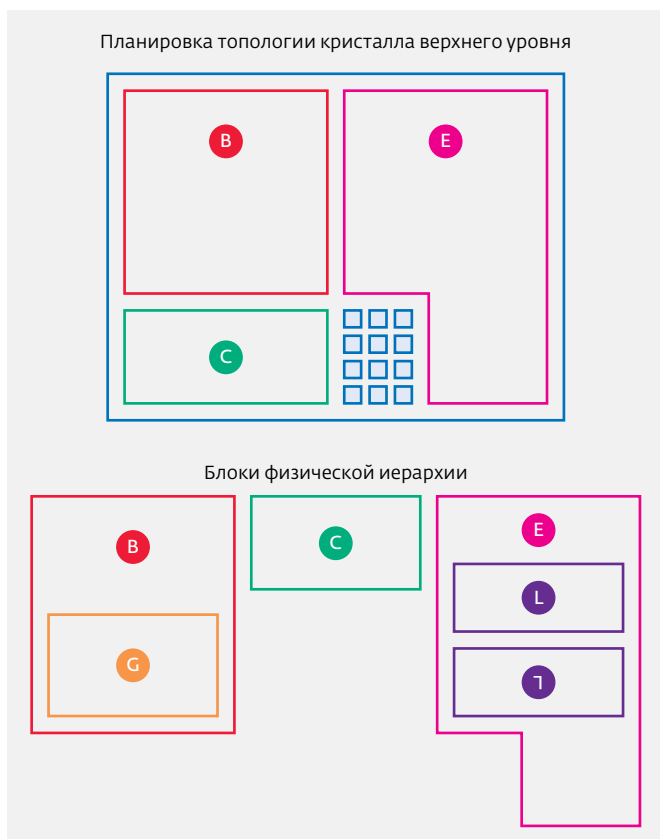


Рис.4. Многоуровневая физическая планировка, разделенная на несколько двухуровневых планировок (верхний уровень и блоки)

ПОДДЕРЖКА МНОГУРОВНЕВОЙ МЕТОДОЛОГИИ ФИЗИЧЕСКОГО ПРОЕКТИРОВАНИЯ КРИСТАЛЛА

Ключ к повышению эффективности работы – поддержка инструментом физического проектирования многоуровневой иерархической структуры проекта. Это означает, что планирование кристалла должно выполняться одновременно на всех уровнях физической иерархии с учетом влияния каждого изменения на другие уровни, блоки и проект в целом. Разработчики должны быть свободны от жестких ограничений более низкого уровня иерархии, которые могут приводить к необходимости сложных согласований проектных решений между разработчиками.

Инструмент физического проектирования должен быть единым, то есть проводить планирование кристалла на всех уровнях и для всех блоков одновременно: автоматически определять нужную форму и размеры блоков, размещать макроячейки (например, блоки памяти) и стандартные ячейки, выполнять трассировку сетки питания, размещать порты входов-выходов, а также генерировать ограничения по временным параметрам (так называемые бюджеты) для всех блоков в контексте всего кристалла. При этом, такой инструмент должен поддер-

живать возможность использования еще не спроектированных блоков, так называемых черных ящиков.

Автоматизированные этапы планировки должны завершаться в течение 12 ч. Разработчики должны иметь возможность запускать задачи иерархической планировки на ночь, а затем в пределах рабочего дня анализировать результаты, изменять глобальные и локальные ограничения и настройки и делать новые запуски. Такой режим работы позволяет максимально повысить эффективность работы над проектом.

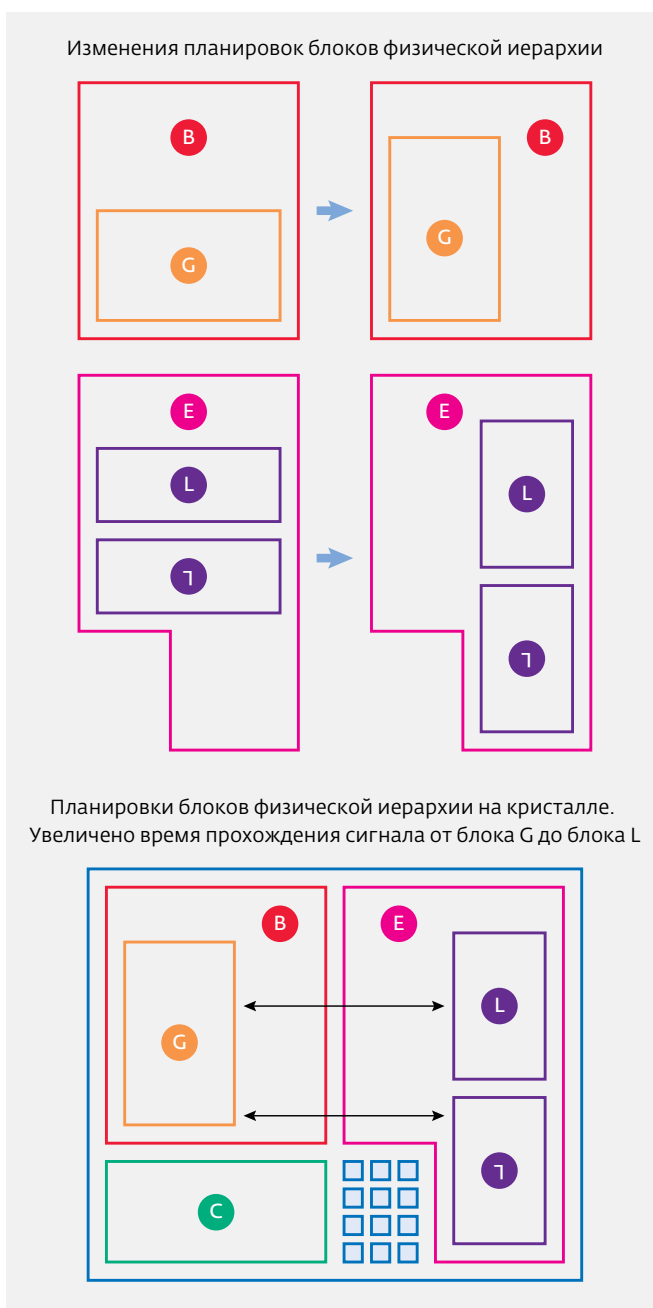


Рис.5. Пример изменений в блоках физической иерархии, которые могут усложнить выполнение верхнего уровня

Процесс планирования кристалла должен максимально использовать данные списков цепей. Сведения о требованиях к занимаемой площади, количестве макроячеек, возможностях реализации трассировки и структурах логических схем на границах блоков позволяют принять оптимальные планировочные решения без необходимости вводить какие-либо предположения (автоматически или инженером). Максимальное использование всех доступных данных, которые влияют на выбор оптимальной планировки, максимально повышает эффективность работы.

Единый маршрут планировки кристалла должен поддерживать любые стили реализации топологии проекта, в том числе и те, в которых между блоками физической иерархии имеются свободные для трассировки каналы (channeled); проекты, где блоки примыкают друг к другу (abutted); проекты с несколькими доменами питания; проекты, содержащие MIB или любую их комбинацию. При этом разработчики должны или сами определить требования к компоновке, например использование блоков физической иерархии и доменов питания, или позволить инструментам проектирования автоматически определять эти требования исходя из заданной площади кристалла. Для реализации стиля abutted необходима поддержка создания сквозных каналов для сигналов, проходящих через блоки, но не использующихся в них (feedthrough). Разработчики должны иметь возможность контролировать создание сквозных каналов трассировки для определенного ими списка цепей и шин кристалла или использовать для этого специальные алгоритмы автоматического создания таких каналов.

Следует отметить, что применение MIB требует особого подхода. Инструмент физического проектирования должен учитывать контекст каждой копии MIB на каждом уровне проекта и корректно определять характеристики эталонной копии (reference), которая позволяет добиться оптимальных характеристик всей системы, то есть будет оптимальной для всех контекстов. К этим характеристикам относится форма блока, координаты размещения портов входов-выходов и бюджетирование. Межблочные соединения для проектов в стиле abutted представляют собой особый случай. Не всегда одни и те же цепи пересекают все копии MIB, не говоря уже об одинаковом их расположении (рис.6). Цепи, назначаемые каналам сквозного соединения, могут быть различными в разных копиях блоков. Кроме того, вероятнее всего

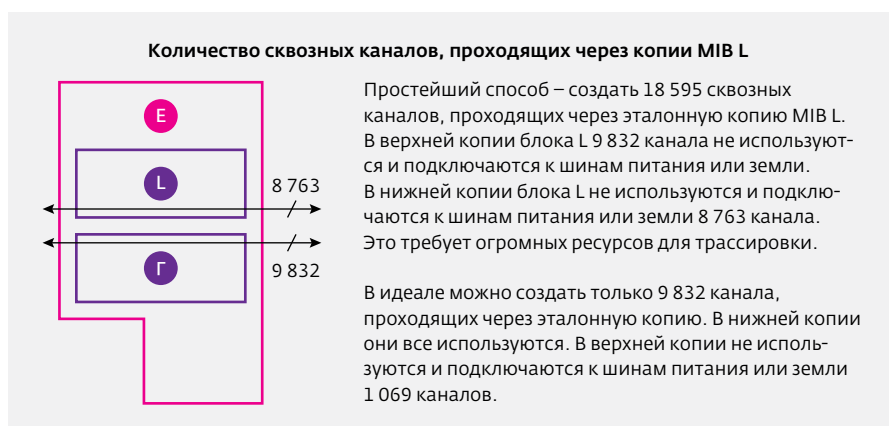


Рис.6. Сквозные каналы, проходящие через блоки MIB

и число сквозных каналов в каждой копии будет разным, хотя, по определению, эталонный блок MIB должен иметь фиксированное число этих каналов. В некоторых копиях блоков используются не все сквозные каналы, поэтому неиспользуемые соединения следует подсоединить к цепям питания или земли в зависимости от конкретной копии MIB.

Важно также, чтобы разработчики обладали возможностью в интерактивном режиме исследовать, анализировать и модифицировать результаты планировки кристалла как на любом отдельном уровне физической иерархии, так и на уровне всего кристалла в целом. Такие возможности позволяют инженерам проводить анализ полученных результатов за минимальное время и достигать оптимального качества планировки кристалла. Кроме того, в этом случае инженеры-топологи могут идентифицировать возникшие проблемы, исправлять ошибки и обеспечивать взаимодействие с разработчиками логического проекта, если необходимо внесение изменений в архитектуру проекта или существенное изменение RTL-описания.

IC Compiler II: НОВЫЙ ПОДХОД К ПЛАНИРОВАНИЮ КРИСТАЛЛА

Ключевой особенностью инструмента топологического проектирования нового поколения IC Compiler II является то, что модель данных имеет структуру, отличную от использовавшихся ранее, которая полностью поддерживает многоуровневую физическую иерархию проекта. Все возможности планирования кристалла инструмента IC Compiler II, среди которых проектирование границ и форм блоков, размещение, трассировка и определение временных ограничений, обеспечивают быстрый доступ к тем данным проекта, которые требуются для реализации конкретной функции.

Инструмент IC Compiler II поддерживает вложенную многоуровневую систему планирования кристалла

с использованием так называемых абстрактных представлений (abstract view) больших блоков в контексте полного проекта, обеспечивая достижение высокого качества результатов и в то же время минимизируя используемую оперативную память. Применение технологии абстрактного представления в IC Compiler II позволяет избежать необходимости загрузки полного списка цепей (netlist) проекта в память в течение всего процесса планирования кристалла и существенно уменьшить затрачиваемое на этот процесс время без потери качества конечного результата.

Процесс создания планировки в инструменте IC Compiler II начинается с разделения проекта на модули и создания физических блоков на основе логической иерархии (partitioning). В некоторых случаях новый проект состоит из ранее использованных блоков, и обычно в такой ситуации логическая иерархия СпК уже создана таким образом, что она прямо соответствует желаемой физической иерархии (блокам). В таких случаях имеющиеся логические модули на любом иерархическом уровне могут быть напрямую преобразованы (commit) в физические блоки. Однако в случае совсем новых проектов инженерам приходится решать отдельную задачу идентификации логических модулей на разных уровнях иерархии, которые должны соответствовать физическим блокам. В этой ситуации часто приходится менять логическую структуру проекта, группируя или разбивая изначальные логические модули в попытке прийти к наиболее оптимальной структуре физической иерархии. Для упрощения решения этой задачи в инструменте IC Compiler II имеется средство Hierarchy Exploration. С его помощью разработчики могут быстро исследовать различные варианты разделения блока на модули и решить, как эти блоки должны быть структурированы логически, чтобы соответствовать желаемой физической иерархии. После того как желаемая физическая иерархия определена, соответствующая информация может быть передана обратно разработчикам RTL для реструктурирования списка цепей, чтобы логическая иерархия соответствовала желаемой физической (блокам).

После задания логической структуры и иерархии проекта на их основе создаются (commit) физические блоки. Дальнейший маршрут планирования кристалла одинаков для любого стиля проектирования (channeled, abutted, MIB и т.д.). В результате преобразования логических модулей в блоки автоматически генерируются представления блоков в виде абстрактов, и после этого процесс планирования "сверху вниз" идет по унифицированному маршруту, который начинается с размещения и определения границ блоков и доменов питания.

Алгоритм, отвечающий за эту функцию, должен учитывать информацию о максимальной площади каждого

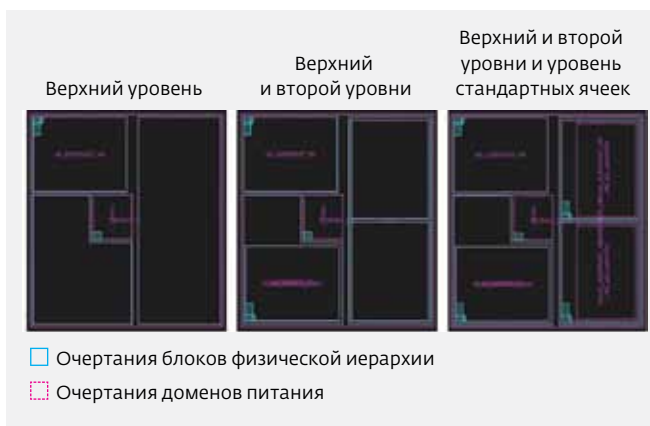


Рис.7. Результаты проектирования форм блоков на нескольких уровнях с помощью инструмента IC Compiler II

блока, соотношении длин сторон блоков (аспектном отношении), которое зависит в том числе от наличия и количества макроячеек, и о соединениях между блоками как одного, так и различных уровней физической иерархии (рис.7). Если в проекте используется несколько доменов питания, то алгоритм дополнительно должен располагать данными о размере доменов питания. При планировании кристаллов с многоуровневой физической иерархией ограничения по форме блоков нижнего уровня передаются на верхний уровень в виде определенных ограничений для блоков верхнего уровня. В то же время алгоритму проектирования формы блоков не нужно обладать информацией о полном списке цепей, существующих в пределах каждого блока. База данных IC Compiler II построена так, чтобы обеспечить все инструменты теми данными, которые нужны для реализации конкретной функции в соответствующий момент времени.

Новая модель данных инструмента IC Compiler II обеспечивает поддержку распределенных процессорных архитектур. Инструмент IC Compiler II распределяет свои задачи или части задач на несколько ядер или процессоров. Хорошим примером такого подхода является процесс размещения макроячеек (рис.8).

После определения границ и форм блоков алгоритм размещения анализирует цепи на границах физической иерархии и определяет необходимые ресурсы трассировки для подключения к макроячейкам. С учетом этой информации выполняется размещение макроячеек для каждого блока на всех уровнях иерархии. Информация о взаимном расположении цепей на границах блоков гарантирует выделение ресурсов, необходимых для трассировки цепей между блоками; инструмент учитывает требования к размещению портов и помещает макроячейки таким образом, чтобы минимизировать

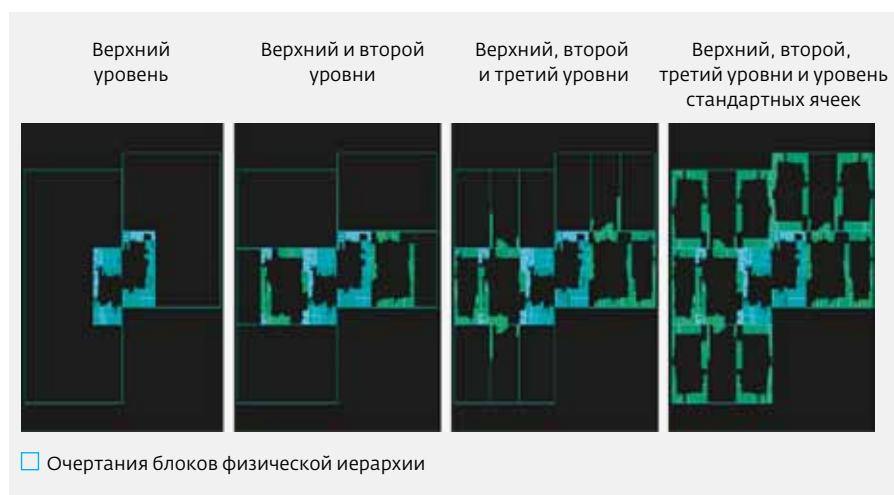


Рис.8. Результаты глобального размещения макроячеек на нескольких уровнях с помощью инструмента IC Compiler II

использование дополнительных усилителей (логических ячеек буферов и инверторов) для передачи сигналов к макроячейкам.

Используя форму и расположение блоков, а также информацию о расположении макроячеек, алгоритм размещения создает параметры внешнего окружения для каждого блока с учетом границ блоков как верхнего,

так и нижнего уровней. Используя эту модель, алгоритм размещения запускает задания по расположению ячеек в каждом блоке на каждом уровне физической иерархии. Распределяя эти задания на множестве процессорных ядер или процессоров, инструмент IC Compiler II минимизирует время завершения задачи и позволяет использовать вычислительные ресурсы максимально эффективно.

Для создания сетки питания в инструменте IC Compiler II реализован инновационный объектно-ориентированный подход. Он основан на использовании шаблонов и стратегий. С помощью шаблонов описывают правила трассировки сетки питания, в том числе ширину проводников, используемые металлические слои, последовательность и шаг проводников. Стратегии определяют, какой шаблон или несколько шаблонов, описывающих конфигурацию сетки питания (одного или нескольких доменов), будут использоваться в определенной пользователем цепи или цепях питания и области кристалла.

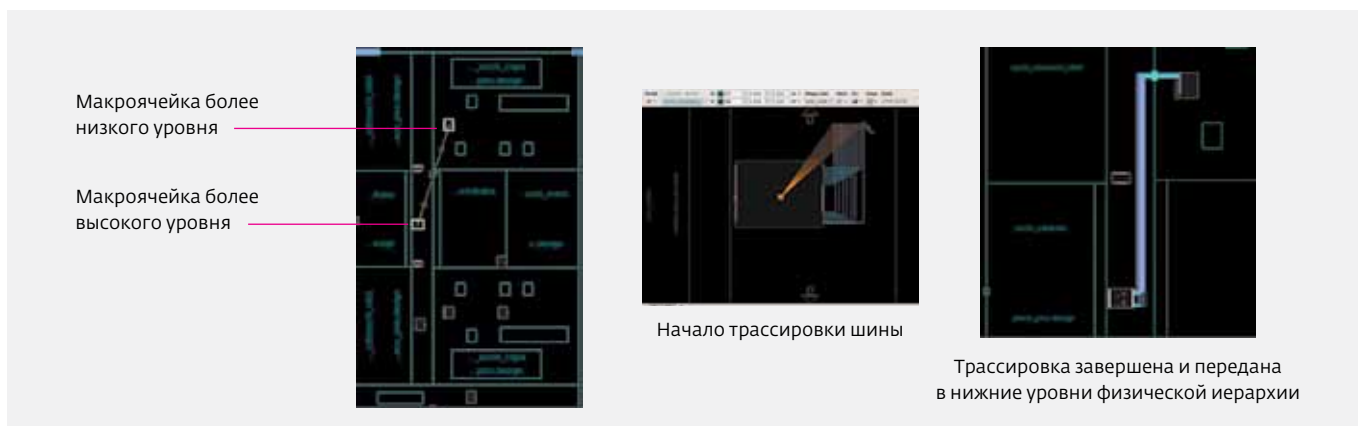


Рис.9. Интерактивное редактирование трассировки на нескольких уровнях

Для всего кристалла может быть создан целый набор стратегий, описывающих конфигурацию питания в различных областях кристалла. Непосредственно создание сетки питания осуществляется с помощью компиляции нужного набора стратегий.

Современные СнК часто включают много шин, соединяющих различные блоки проекта. Для планирования топологии таких шин инструмент IC Compiler II предлагает набор средств, позволяющих выполнять данную задачу на верхнем уровне физической иерархии с последующей передачей информации о проложенных шинах блокам нижних уровней (рис.9). Это включает в себя, в частности, возможность планирования топологии шины, используя интерактивный способ, где инженеры могут вручную проводить детальную трассировку шины. Этот способ планирования шины дает разработчику полный контроль над ее топологией. Другой способ, поддерживаемый в IC Compiler II, – это планиро-

вание топологии шины с помощью создания ограничений, например создание коридоров для трассировки, а сама трассировка шин в таком случае осуществляется с помощью алгоритма глобальной трассировки. Второй способ обычно является предпочтительным, если возможны значительные изменения в планировке кристалла. Оба метода поддерживают планирование шин в проектах с многоуровневой вложенной физической иерархией.

После того как определены формы блоков, размещены макроячейки и выполнена компиляция сетки питания, начинает работать алгоритм размещения портов, входящий в состав инструмента IC Compiler II. Он извлекает данные о границах каждого блока на всех уровнях физической иерархии и запускает алгоритм глобальной трассировки сигнальных цепей для того, чтобы определить точные координаты размещения иерархических портов. Алгоритм глобальной трассировки учитывает имеющиеся в проекте MIB, что гарантирует одинаковое пересечение цепями границ всех MIB. Специальный алгоритм определяет наилучшее расположение портов, подходящее для всех копий MIB, и следит за тем, чтобы размещение портов в каждой копии MIB было идентичным. Кроме того, при размещении портов создаются цепи для сквозных сигналов (feedthrough) для всех блоков, включая MIB, по всей иерархии (рис.10). Глобальная трассировка планирует сквозную трассировку через MIB и определяет, какие неиспользуемые цепи сквозных сигналов следует подсоединить к цепям питания или земли. При этом наи-

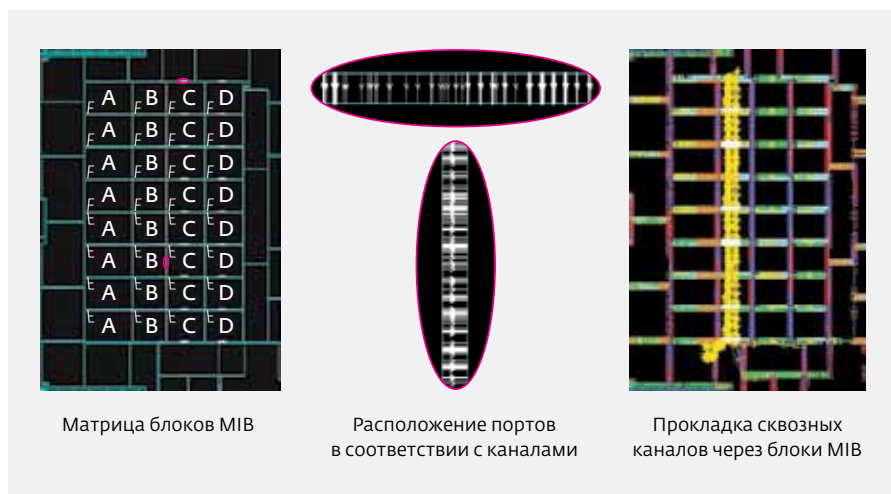


Рис.10. Результаты привязки портов и создания сквозных каналов с помощью инструмента IC Compiler II

более важным аспектом работы с MIB является то, что разработчику нет необходимости определять эталонную копию блока, чтобы повысить качество результатов: все операции внутри маршрутов планировки в инструменте IC Compiler II созданы таким образом, чтобы находить решение, оптимальное одновременно для всех копий MIB.

После размещения иерархических портов на основании полученных при глобальной трассировке параметров сопротивления и емкости (RC) для каждой цепи инструмент IC Compiler II выполняет расчет параметров быстродействия сигналов, проходящих через границы блоков физической иерархии, и создает набор временных ограничений (бюджеты) для каждого блока. Большинство инструментов проектирования создает ограничения по временным параметрам для входных и выходных портов блоков физической иерархии. Бюджеты отражают параметры быстродействия межблочных соединений. Это дает возможность инженерам-разработчикам блоков, входящих в состав кристалла, выполнять размещение и оптимизацию своих блоков путем аннотации параметров быстродействия, полученных на этапе планирования кристалла. Однако это не позволяет рассчитывать параметры быстродействия внутри блоков с точки зрения блоков более высокого уровня иерархии. Усовершенствованный алгоритм расчета параметров быстродействия в составе инструмента IC Compiler II позволяет одновременно создавать бюджеты для блоков всех уровней иерархии. Это позволяет реализовать параллельное размещение портов и его оптимизацию в рамках всего проекта одновременно.

В процессе разработки планировки кристалла инженеры-топологи имеют возможность определять наборы модифицируемых и немодифицируемых блоков физической иерархии. Эта возможность может помочь, когда

нужно обновить один из блоков в силу того, что его состав подвергся изменению. Если исходя из нового списка цепей требуется, например, увеличить размер блока, то инструмент IC Compiler II в пошаговом режиме обновляет планировку, учитывая контекст всего кристалла, а также наборы модифицируемых блоков. Это позволяет минимизировать время, которое требуется разработчикам для согласования необходимых изменений на разных уровнях иерархии кристалла.

На любом этапе планирования и создания топологии кристалла разработчики могут в интерактивном режиме анализировать и вручную редактировать любой уровень физической иерархии в контексте всего кристалла. Например, разработчики могут вручную выполнить трассировку критичной с точки зрения временных параметров сигнальной цепи, проходящей через несколько уровней иерархии. В инструменте IC Compiler II они могут открыть полный проект в окне редактирования топологии и выбрать отображение только верхнего уровня или нескольких уровней иерархии. При отображении нескольких уровней интерактивная трассировка выполняется так, как если бы проект был плоским. По завершении процесса трассировки помещается в блоки и в них автоматически добавляются новые иерархические порты.

Как было отмечено выше, инструмент IC Compiler II также включает в себя богатый набор инновационных возможностей для создания и анализа качества результатов по планировке, которые могут использоваться на протяжении всего маршрута пошагово. Это дает разработчику возможность оценивать качество планировки на каждом этапе и вырабатывать стратегию улучшения качества результатов по мере продвижения по маршруту. Все перечисленные возможности поддерживают многоуровневую физическую структуру проекта. ●