

## **Решим, какие блоки проекта будут его разделами**

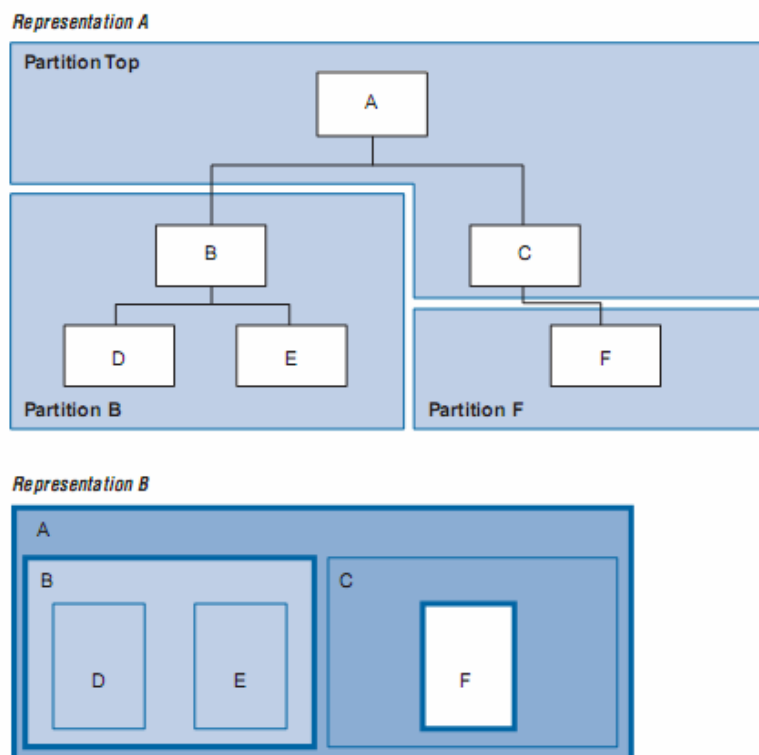
Общей практикой создания проектов является их модульная или иерархическая архитектура, при которой вы разрабатываете отдельно каждый блок, а затем компонуете из таких блоков целый проект. Программа Quartus II изначально не воспринимает каждый блок в качестве раздела проекта, для его использования в процессе инкрементной компиляции, следовательно, вам необходимо разработать одну или несколько иерархий проекта ниже раздела головного модуля проекта. Создание разделов мешает компилятору оптимизировать связи между разделами, как это описано в главе "Влияние разделов проекта на оптимизацию проекта" на странице 2-11. Таким образом, возможен независимый синтез и размещение каждого отдельного раздела, что делает возможным инкрементную компиляцию.

Разделы должны иметь те же самые границы, что и иерархические блоки в проекте, поэтому недопустимо, чтобы раздел содержал какую-либо часть логики из иерархического блока. Когда вы объявили раздел, каждый иерархический модуль внутри становится частью этого раздела. Вы можете создать новые разделы для иерархических блоков внутри существующего раздела, в таком случае, блоки внутри нового раздела уже не являются частью раздела более верхнего уровня, как это описано в следующем примере.

На рисунке 2-4 иерархические блоки В и F формируют разделы в законченном проекте, который состоит из блоков А, В, С, D, Е и F. Заштрихованные области на изображении А показывают разделы проекта в виде иерархического "древа". На изображении В низкоуровневые

блоки показаны внутри высокоуровневых блоков, а разделы показаны различной штриховкой. Головной раздел, называемый TOP, автоматически содержит блоки верхнего уровня и прочую логику, не относящуюся ни к какому другому разделу. Файл проекта верхнего уровня – это упаковщик для иерархической структуры нижних уровней, он может также содержать свою логику. В этом примере, раздел для блока верхнего уровня А также включает в себя логику одного блока нижнего уровня, С. Поэтому, блок F содержится в своем собственном разделе, он не является частью раздела верхнего уровня. Другой отдельный раздел В содержит в себе логику блоков В, D и E.

**Figure 2-4. Partitions in a Hierarchical Design**



Вы можете назначить раздел любому блоку проекта. Блоки могут быть описаны в коде HDL или схематически, или получены инструментом синтеза сторонних разработчиков VQM или быть блоком списка соединений EDIF.

Для использования преимуществ инкрементной компиляции при изменении файла исходного кода, создавайте отдельные файлы проектов для каждого раздела. Если два различных блока в различных разделах, но объединены одним файлом проекта, вы не сможете использовать преимущества инкрементной компиляции, потому что программа будет перекомпилировать оба раздела, если вы измените что-либо в файле проекта. Аналогично, если два раздела зависят от описания одного низкоуровневого блока, изменения в нём повлияет на оба раздела.

В заключении этой главы содержится информация, которая поможет вам выбрать, какие блоки проекта назначить его разделами.

### **Влияние разделов проекта на оптимизацию проекта**

Границы разделов вашего проекта оказывают влияние на качественные характеристики проектов. Создание разделов мешает компилятору проводить логическую оптимизацию между границами разделов, позволяя программе в процессе инкрементной компиляции проводить синтез и размещение каждого раздела независимо. Таким образом, необходимо следовать руководству по созданию разделов для того, чтобы уменьшить негативный эффект границ разделов.

Насколько возможно, регистрируйте все выходы и входы для каждого раздела. Это поможет избежать некоторых проблем с задержками сигналов между границами разделов и сохранит для оптимизации каждый временной путь от регистра к регистру внутри одного раздела. К тому же, уменьшит количество путей между границами разделов. Если критичные временные пути располагаются между границами разделов, необходимо переработать раздел, чтобы эти пути не выходили из раздела. Включение всех возможных критических путей внутрь раздела позволяет эффективно применять оптимизацию к разделу для улучшения временных характеристик, когда остальная часть проекта не изменяется. Дополнительно, избегайте констант на входах и выходах раздела, потому что для поддержания инкрементных свойств, программа не сможет использовать константы для оптимизации логики на любой стороне границы раздела.

Планировщик раздела проекта поможет вам сделать хорошие назначения, как это описано в главе "Создание назначений раздела проекта" на странице 2-16. В следующих главах описаны инструменты, которые вы можете использовать после компиляции для анализа назначений проекта. Вы можете просматривать "Отчёт статистики разделов", содержащий информацию о количестве I/O соединений и о том, какие из них не зарегистрированные или подключенные к константе, в виде отчётов статистики разделов. Вы можете также создать "Временные отчёты разделов" и обратиться к "Советнику по инкрементной компиляции" для анализа и поддержки.

Если между границами разделов находятся критичные временные пути, вы можете выполнить временное планирование и создать временные назначения, чтобы обязать логику внутри каждого раздела иметь соответствующие требованиям временные пути. К тому же, когда каждый раздел оптимизируется во время синтеза независимо друг от друга, вы можете сделать балансировку ресурсов, чтобы добиться того, чтобы каждый раздел использовал только определённое вами количество ресурсов чипа. В процессе восходящей компиляции, когда разделы проекта компилируются независимо от головного проекта, могут возникнуть конфликты в использовании глобальных ресурсов разводки тактовых сигналов, во время импорта раздела проекта. Вам необходимо использовать мегафункцию ALTCLK\_CTRL, чтобы разместить блоки контроля тактов и подключить их соответственно в нижнем и верхнем уровнях проекта, или найти сгенерированные компилятором узел контроля тактов в вашем проекте, и создать назначения локализации контроля тактов в Редакторе Назначений.

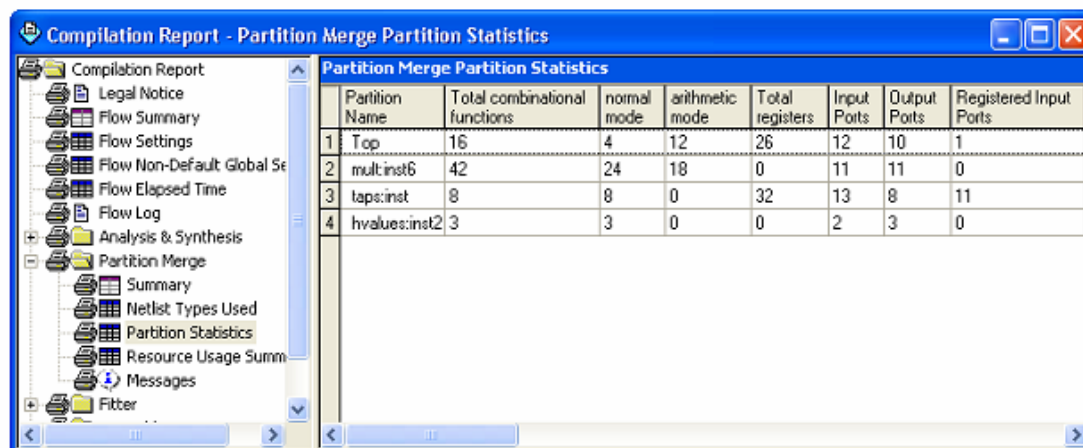
За дополнительными советами и специальными рекомендациями для закрепления основных характеристик проекта, таких как информация о балансировке ресурсов и временное планирование, обратитесь к главе "Лучшие примеры для инкрементной компиляции разделов и назначений в Архитектуре" в томе 1 "Настольной книги Quartus II".

## **Отчёты статистики разделов**

После компиляции, вы можете наблюдать статистику разделов проекта в отчете Статистики объединенных разделов или на вкладке **Статистика** в диалоговом окне **Свойства разделов проекта**.

Страница **Статистики раздела** под папкой **Объединение разделов Отчета о компиляции** показывает статистику каждого раздела. Статистика каждого раздела (каждая строка в таблице) содержит число содержащихся в нем логических ячеек, а также число входных и выходных выводов, и сколько из них зарегистрировано или не подключено. Этот отчет полезен, когда вы оптимизируете проект, в процессе нисходящей компиляции, или во время компиляции головного проекта в восходящем процессе компиляции, позволяя проверить, насколько раздел соответствует рекомендациям, сделанным в главе "Лучшие примеры для инкрементной компиляции разделов и назначений в Архитектуре" в томе 1 "Настольной книги QuartusII". На рисунке 2-5 показано окно отчета.

**Figure 2-5.** Partition Merge Partition Statistics Report



	Partition Name	Total combinational functions	normal mode	arithmetic mode	Total registers	Input Ports	Output Ports	Registered Input Ports
1	Top	16	4	12	26	12	10	1
2	mult:inst6	42	24	18	0	11	11	0
3	taps:inst	8	8	0	32	13	8	11
4	hvalues:inst2	3	3	0	0	2	3	0

Вы можете также наблюдать статистику ресурсов и соединений портов для каждого отдельного раздела на вкладке **Статистика** в диалоговом окне **Свойства разделов проекта**. В меню **Назначения**, кликните **Окно Разделов проекта**. Правым кликом на раздел и кликом на **Свойства** для открытия диалогового окна. Выберите **Показать все разделы** для того, чтобы просмотреть все разделы в этом отчете (рисунок 2-6).

**Figure 2-6. Statistics Tab in the Design Partitions Properties Dialog Box**

Design Partition Properties -- hvalues:inst2

General | Compilation | Statistics

Displays the post-compilation statistics for the design partitions selected in the Design Partitions window.

Statistic	Top	hvalues:inst2	mult:inst6	taps:inst
<b>Resources</b>				
--- I/O	15	0	0	8
--- Combinational cell	22	3	42	12
--- Register cell	26	0	0	32
--- Clock control block	3	0	0	0
<b>Connections</b>				
--- Input Connections	11	2	48	112
--- Registered Input Connections	0	0	0	96
--- Output Connections	114	24	11	24
--- Registered Output Connections	0	0	0	0
<b>Internal Congestion</b>				
--- Total Connections	254	26	106	204
--- Registered Connections	62	0	0	152
<b>Inter-partition connections</b>				
--- Top	0	2	11	112
--- hvalues:inst2	2	0	24	0
--- mult:inst6	11	24	0	24
--- taps:inst	112	0	24	0

Show All Partitions

OK Cancel Apply

## Временные отчёты разделов

Вы можете генерировать Общий временной отчёт раздела и Детальный временной отчёт, если кликните на **Отчёт разделов** в панели Задачи во временном анализаторе TimeQuest или воспользовавшись Tcl командой `report_partitions`.

В Общем временном отчёте раздела показано общее количество неудачных путей для каждого раздела и наихудший временной резерв для некоторого количества путей внутри раздела.

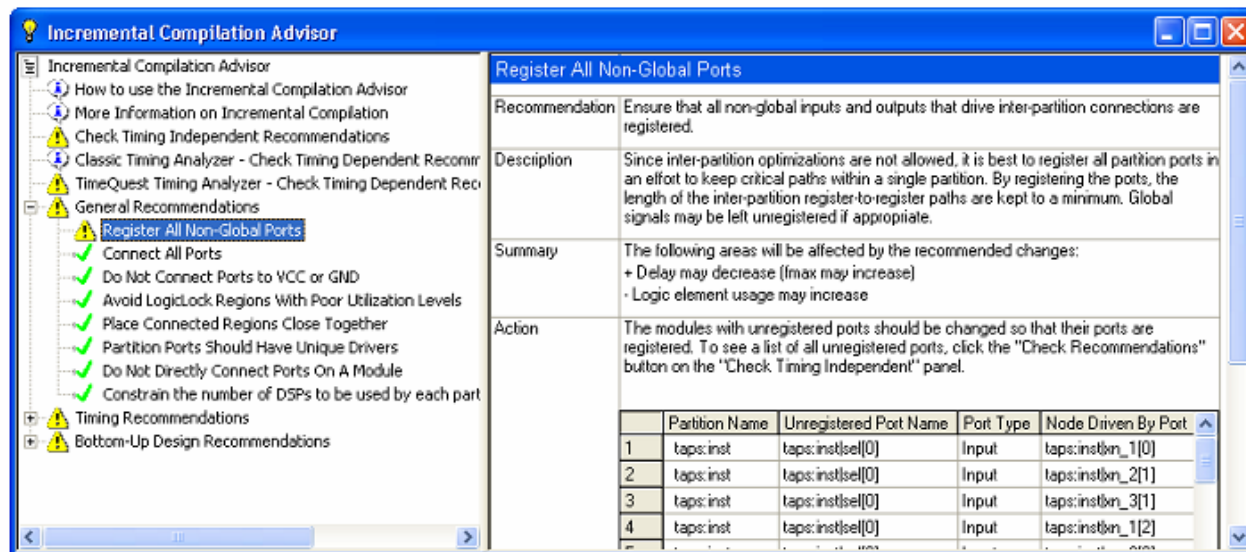
В Детальном временном отчёте показано количество неудачных путей от раздела к разделу и наихудший временной резерв для путей от раздела к разделу, чтобы уточнить, какие критические пути в проекте отдельно от разделов проекта могут вывести из строя весь проект.

## Советник по инкрементной компиляции

Вы можете использовать Советник по инкрементной компиляции для проверки того, насколько ваш проект сделан по рекомендациям Altera для создания разделов и назначений локализации в Архитектуре. В меню Инструменты, пункт **Советник**, выберите **Советник по инкрементной компиляции**.

Как показано на рисунке 2-7, рекомендации делятся на **Главные Рекомендации**, которые применимы ко всем процессам компиляции и **Рекомендациям для Восходящего проектирования**, которые применяются в методе восходящего проектирования. Каждая рекомендация содержит разъяснения, которые описывают эффект от рекомендаций, и порядок действий, для внесения изменений. В некоторых случаях, это ссылка на соответствующую страницу настройки программы Quartus II, на которой вносятся необходимые изменения в назначения или настройки. Различные независимые от времени рекомендации, также находятся в Окне Разделов проекта и в Окне регионов LogicLock.

**Figure 2-7. Incremental Compilation Advisor**



Для проверки в будущем, следует ли проект рекомендациям, обратитесь к странице **Зависимые от времени рекомендации** или к странице **Независимые от времени рекомендации**, и кликните **Сверить рекомендации**. Для больших проектов, эти операции занимают несколько минут. После того, как вы сделаете операцию сверки, появятся символы следования рекомендации, как показано на рисунке 2-8, показывающие, насколько настройки проекта следуют или не следуют рекомендациям. Обратитесь к легенде на странице **Как пользоваться советником по инкрементной компиляции** в Советнике за дополнительной информацией.

Кроме того, в Советнике, даже если ваш проект не следует рекомендациям, то при операции **Сверить рекомендации** отображают некоторые части проекта, которые могут быть улучшены. Например, если не все порты I/O разделов следуют рекомендации **Регистрировать все порты**, советник покажет список не зарегистрированных портов с именами разделов и именами узлов, ассоциированных с этим портом.

Когда советник подготовил список узлов, вы сможете правым кликом на узел выбрать **Локализовать**, чтобы получить межпробник, использующийся в других средствах Quartus II, таких как RTL Просмотрщик, Планировщик чипа, или текстовых исходных кодах в текстовом редакторе.

Открытие нового отчёта TimeQuest сбрасывает результаты Советника по инкрементной компиляции, поэтому вам необходимо перезапустить процесс Проверки рекомендаций.

## **Использование разделов с помощью инструментов синтеза сторонних разработчиков**

Если вы используете инструмент синтеза сторонних разработчиков, настройте ваши инструменты для создания отдельных VQM или EDIF списков соединений для каждого иерархического раздела. В программе Quartus II, в головном проекте назначьте каждому списку соединений раздел проекта. VQM или EDIF списки соединений в программе Quartus II обрабатываются как исходный файл для раздела.

### **Synplicity Synplify Pro/Premier и Mentor Graphics Precision RTL Plus**

Программы Synplify Pro и Synplify Premier содержат средство синтеза MultiPoint для реализации инкрементного синтеза каждого блока проекта, обозначенного как Точка компиляции в пользовательском интерфейсе или с помощью скрипта. Программа Precision RTL Plus содержит средство инкрементного синтеза, который реализуется поблочным синтезом, основанным на назначениях к разделу в исходном HDL коде. Эти средства проводят автоматический поблочный инкрементный синтез и создают различные выходные файлы списков соединений для каждого блока, определённого для чипа Altera.

Использование инструмента инкрементного синтеза гарантирует, что только те части проекта, которые были обновлены, пересинтезируются во время компиляции проекта, что уменьшает время синтеза и сохраняет результаты для неизменяемых блоков. Вы можете изменить и пересинтезировать одну часть проекта, не затрагивая других его частей.

За более подробной информацией об этих процессах инкрементного синтеза обратитесь к документации поставщика инструментария, или к соответствующей главе в томе 1 "Настольной книги Quartus II: Поддержка Synopsys Synplify или синтеза Mentor Graphics Precision".

### **Другие инструментарии синтеза**

Вы можете также разделять ваш проект и создавать различные файлы списков соединений вручную с базовой программой Synplify (не Pro/Premier), базовой программой Precision RTL (не Plus), или любым другим поддерживаемым инструментом синтеза, путём создания различных проектов или реализации любого раздела, включая головной. Задайте в каждом проекте верхнего уровня, блоки, состоящие из низкоуровневых списков соединений VQM/EDIF в виде чёрных ящиков. Synplify, Precision и большинство инструментов синтеза автоматически обрабатывают блок проекта как чёрный ящик, даже если логические обозначения отсутствуют в проекте. Каждый инструмент обладает опциями и атрибутами, чтобы выбрать, какой блок проекта сопоставляется с чёрным ящиком, что позволит вам избежать предупреждений о потерянной логике.

## **Сравнение Назначений разделов проекта и Назначений физического размещения**

Разделы проекта для инкрементной компиляции являются логическими разделами, их следует отличать от физического размещения, назначенного в Архитектуре чипа. Логический раздел не связан с физическим местом на чипе и, поэтому, не может контролировать размещение модулей. Логический раздел проекта устанавливает виртуальные границы между иерархиями проекта, так чтобы каждый отдельный блок компилировался независимо, препятствуя логической оптимизации между блоками. Когда программа компилирует исходный код проекта, логика в каждом разделе может размещаться в любом месте на чипе, если вы не сделаете дополнительные назначения. Программа создает отдельные пост-синтез и пост-компоновка списки соединений для

каждого раздела, которые используются для повторения результатов синтеза и компоновки (включая информацию о размещении и разводке) в последующей компиляции.

Если вы хотите сохранить результаты компиляции, используя список соединений пост-компоновка, то вам не потребуется обратная аннотация или создание каких-либо назначения локализации для определённых логических узлов. Вы не сможете использовать одновременно инкрементную компиляцию и средство назначения обратной аннотации в одном проекте Quartus II. Средство инкрементная компиляция не использует размещение "назначений" для сохранения результатов размещения; она только использует повторно базу данных списка соединений, включающую информацию о размещении.

Вы можете назначить разделам проекта физическое место в Архитектуре чипа, используя назначения регионов LogicLock. В программе Quartus II, регионы LogicLock используются для сопоставления блокам проекта индивидуальных регионам на чипе. Altera рекомендует использовать регионы LogicLock для достижения качественных характеристик и избежания конфликтов размещения, во время выполнения инкрементной компиляции. Создание назначений размещения в Архитектуре для разделов проекта с использованием регионов LogicLock, описано в главе "Создание Архитектуры проекта с назначением локализации LogicLock" на странице 2-25.

За более подробной информацией о том, как создавать Архитектуру проекта, обратитесь к главе "Лучшие примеры разделов инкрементной компиляции и назначений архитектуры" в томе 1 "Настольной книги Quartus II".