

## Решим, стоит ли использовать процесс инкрементной компиляции

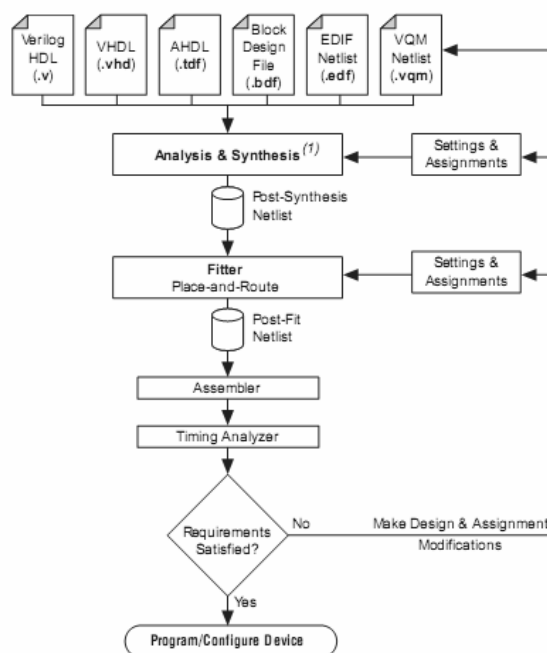
Инкрементная компиляция в Quartus II улучшает стандартный процесс проектирования в Quartus II, позволяя вам сохранять удовлетворительные результаты компиляции и характеристики неизменённых частей. В этой главе описан линейный процесс компиляции проекта без разделов и инкрементный процесс, при котором проект разделен на разделы, и объяснены различия. В этой главе также объяснено, когда лучше использовать линейную компиляцию, и объяснена причина, по которой необходимо создавать разделы и использовать инкрементную компиляцию.

В программе Quartus II опция полной инкрементной компиляции включена по умолчанию, т.е. проект готов для создания разделов и инкрементной компиляции. Если вы не создали разделов проекта, то компилятор будет использовать линейный процесс компиляции, а вы не сможете совершать инкрементные изменения в проекте. Синтез и компоновка проекта без разделов – это то же самое, что и компиляция с выключенной опцией полной инкрементной компиляции Quartus II. Поэтому, Altera рекомендует включить эту опцию, поскольку это позволит вам использовать инкрементные процессы, такие как, отладка с помощью встроенного логического анализатора SignalTap, даже если вы не создали разделов проекта.

### *Линейный процесс компиляции без создания разделов*

На рисунке 2-1 показан процесс компиляции без создания разделов проекта

**Figure 2-1.** Quartus II Compilation Flow with No Design Partitions



Примечания к рисунку 2-1. Если вы используете список соединений EDIF или VQM, созданный инструментами синтеза EDA сторонних разработчиков, Анализ и Синтез создает базу данных проекта, но логический синтез и технологическая разводка осуществляется только для алгоритмических схем и мегафункций Altera.

В процессе (по умолчанию) линейной компиляции без разделов проекта, весь исходный код проходит через Анализ и Синтез, и вся логика размещается и разводится Компоновщиком каждый раз, когда происходит перекомпиляция проекта при внесении изменений в какой-нибудь его части. Одним из оснований такого процесса является получение оптимальных характеристик. Используя

в своем процессе целый проект, компилятор задействует глобальную оптимизацию для улучшения размещения и характеристик.

Вы можете использовать линейную компиляцию для небольших проектов, выполненных на CPLD или небольших FPGA, когда временные ограничения достигаются проще за одну компиляцию. Линейная компиляция оправдана, когда время компиляции и сохранения результатов для временных ограничений не важны.

## **Интеллектуальная компиляция Quartus II**

Программа Quartus II также содержит средство, называемое Интеллектуальная компиляция, которая не надо путать с инкрементной компиляцией. В некоторых процессах компиляции Quartus II, вы можете использовать Разумную компиляцию, чтобы позволить компилятору определить требуемые стадии компиляции, основываясь на изменениях, сделанных в проекте с момента последней разумной компиляции, и пропустить некоторые ненужные стадии. Например, когда включена Интеллектуальная компиляция, компилятор пропускает часть Анализ и Синтез, если все исходные файлы проекта не были изменены. Интеллектуальная компиляция пропускает только целые стадии компиляции. Она не может создавать инкрементные изменения внутри одной стадии процесса компиляции. Для включения Интеллектуальной компиляции, в меню **Назначения**, кликните **Настройки**. В списке **Категории**, выберите **Настройки процесса компиляции** и кликните на **Использовать Интеллектуальную Компиляцию**.

## ***Процесс инкрементной компиляции с разделами проекта***

Использование разделов проекта позволяет вам сохранять результаты и характеристики неизменной логики, когда вы вносите изменения кое-где в проекте, и сократить время компиляции. Инкрементная компиляция рекомендована для больших проектов и высокой концентрации ресурсов, таких как проекты, которым требуются высокие характеристики по отношению к скорости архитектуры чипа. Это средство также задействуется в командных проектах, позволяя разработчикам создавать и оптимизировать свои блоки независимо друг от друга.

Инкрементная компиляция поддерживает нисходящую инкрементную компиляцию, при которой один разработчик управляет одним проектом, и восходящую инкрементную компиляцию, при которой каждый блок проекта разрабатывается и оптимизируется независимо, а также комбинацию обеих компиляций. Посмотрите "Нисходящая версия процесса восходящей компиляции" на странице 2-6. Чтобы воспользоваться преимуществами процесса инкрементной компиляции, начните с разделения проекта по его границам иерархии на блоки, называемые разделами проекта. Обратитесь к "Решим, какие блоки проекта будут его разделами" на странице 2-9, и к "Создание назначений для раздела проекта" на странице 2-16. Программа Quartus II синтезирует отдельно каждый иерархический раздел проекта, затем объединяет их в окончательный список соединений для следующих стадий процесса компиляции. При перекомпиляции проекта, вы можете использовать исходный код, результаты пост-синтеза или пост-компоновки для каждого раздела. Если вы хотите зафиксировать результаты компоновки, вам нужно сохранить результаты размещения, или результаты разводки и размещения.

Вы можете использовать инкрементную компиляцию по окончании проектирования, когда вы не будете улучшать большую часть проекта, а всего лишь захотите внести изменения или оптимизировать один определённый блок. В этом случае, вы можете сохранить характеристики модулей, удовлетворяющие требованиям, чтобы упростить составление временных ограничений и сократить время компиляции при следующих итерациях. Вы можете перекомпилировать другие

разделы с включенной функцией расширенной оптимизации, для улучшения их характеристик без изменений сохранённых разделов.

Часть вашего проекта может быть не полной или разрабатываться различными разработчиками или поставляться IP. Вы можете создать пустой раздел для этой части проекта во время компиляции целых разделов, и тем самым, сохранить результаты для целых разделов во время импорта части проекта. Альтернативно, различные разработчики или поставщики IP могут разрабатывать и оптимизировать различные блоки проекта независимо, а вы сможете комбинировать их в процессе восходящей компиляции.

За большей информацией и примерами рекомендованных процессов разработки для выгодного использования инкрементной компиляции обратитесь к "Рекомендованный процесс проектирования и примеры компиляции" на странице 2-49.

В таблице 2–1 показано кратко влияние инкрементной компиляции на ваши результаты компиляции.

**Таблица 2-1.** Кратко о влиянии использования инкрементной компиляции

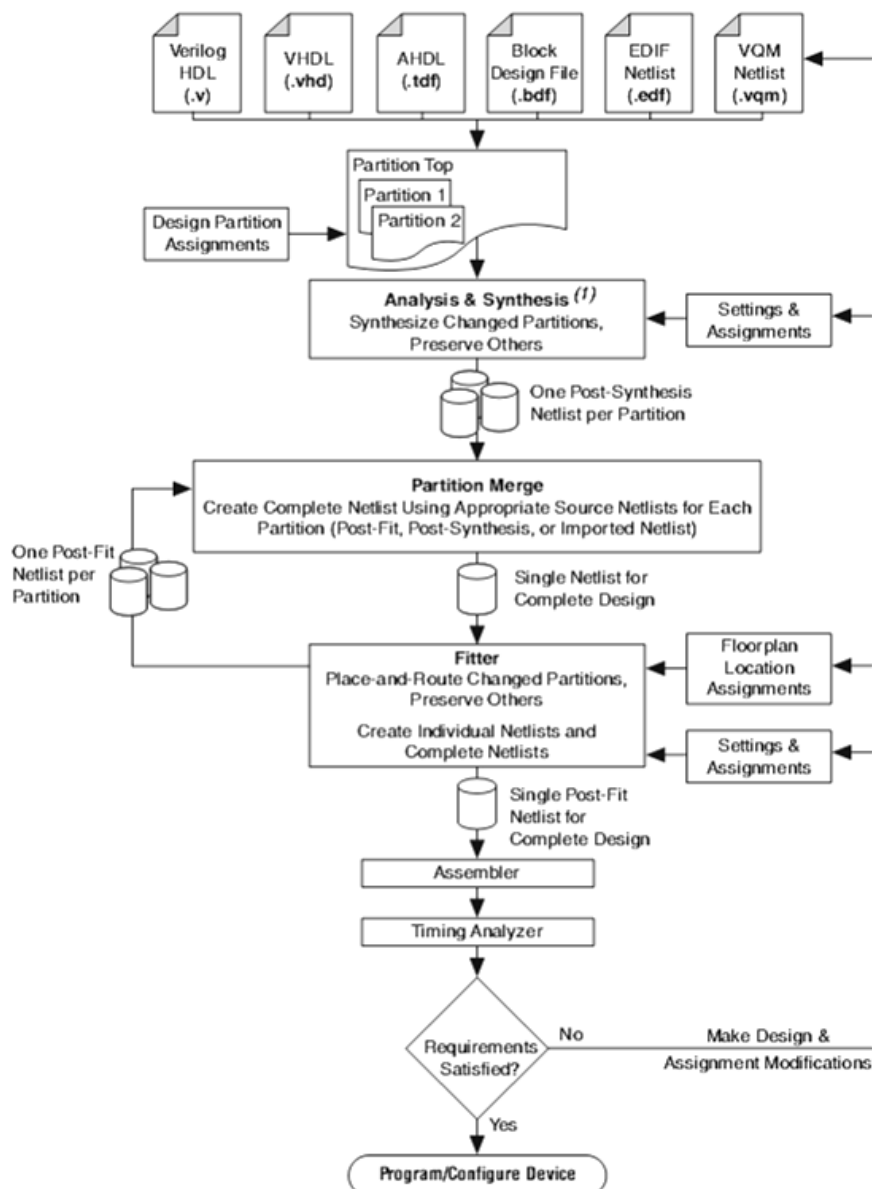
Характеристика	Влияние инкрементной компиляции с разделами проекта
Сохранение времени компиляции	Обычно сохраняется 50-70% времени компиляции, когда сохраняется список соединений пост-компоновка; это сохраняется и в интегрированном синтезе Quartus II и в Компоновщике
Сохранение характеристик	Превосходно, если критические пути находятся внутри раздела, поскольку вы сохраняете информацию пост-компоновка для неизмененных разделов
Сохранение имён узлов	Сохраняет пост-компоновка имена узлов для неизменённых разделов
Изменение площади	Площадь (использования логических ресурсов) может увеличиться, поскольку межграничная оптимизация не возможна, а размещение и упаковка регистров ограничена.
Изменение Fmax	Максимальная частота работы может уменьшиться, поскольку межграничная оптимизация не возможна. Если проект разделен, и назначения локализации архитектуры созданы правильно, то это не окажет негативного влияния на Fmax.
Создание архитектуры	Рекомендовано для критичных разделов для получения более качественных результатов при внесении изменения в проект. Требуется во время импорта разделов из другого проекта Quartus II, чтобы избежать конфликтов размещения.

Если вы используете средство инкрементной компиляции в любой точке вашего процесса проектирования, вам необходимо начать планирование инкрементной компиляции с начала вашей разработки. Так будет проще подогнать границы разделов и создать архитектуру, если вы начинаете планирование с начала вашего цикла разработки.

Обратитесь к "Лучшие примеры разделов инкрементной компиляции и назначений архитектуры" в томе 1 "Настольной книги Quartus II".

На рисунке 2-2 показан процесс разработки с использованием инкрементной компиляции и разделов проекта.

**Figure 2-2. Quartus II Design Flow Using Incremental Compilation**



Примечания к рисунку 2-2.

Если вы используете список соединений EDIF или VQM, созданный инструментами синтеза EDA сторонних разработчиков, Анализ и Синтез создает базу данных проекта, но логический синтез и технологическая разводка осуществляется только для алгоритмических схем и мегафункций Altera.

На диаграмме рисунка 2-2 показан головной раздел с двумя разделами нижнего уровня. Если изменяется какая-либо часть проекта, Анализ и Синтез обрабатывает измененный раздел, оставляя без изменений списки соединений неизмененных разделов. После завершения Анализа и Синтеза, здесь будет один список соединений пост-синтеза для каждого раздела.

Следующим шагом идет объединение разделов в единственный, законченный список соединений, полученный из списков соединений пост-синтеза, списков соединений пост-компоновки, и импортированных списков соединений из других Quartus II проектов, в зависимости от типа списка соединений, определенного для каждого раздела.

Компоновщик обрабатывает объединенный список соединений, фиксирует размещение или размещение с разводкой для неизмененных разделов, и перекомпоновывает только те разделы, которые были изменены. Компоновщик генерирует конечный список соединений для дальнейшего использования на следующих этапах процесса компиляции, включающих временной анализ и генерацию файлов программирования. Т.о., генерируются индивидуальные списки соединений для каждого раздела для того, чтобы на этапе Объединения разделов использовать список соединений пост-компоновка для фиксации размещения и разводки раздела, но тот случай, если эта информация потребуется в дальнейших компиляциях. Программа Quartus II заново не синтезирует и не разводит заново не изменённые разделы, которые имеют назначения типа списка соединения соответственно пост-синтез или пост-компоновка.

За более подробной информацией использования таких свойств инкрементной компиляции обратитесь к "Руководству по быстрому старту – общий подход к процессу инкрементной компиляции" на странице 2-7.

Программа Quartus II ранних версий (до 8.1) имела опцию "только инкрементный синтез", которая не могла сохранять результаты размещения. Эта опция была удалена, начиная с версии 8.1. Вы можете использовать список соединений пост-синтеза для сохранения результатов синтеза во время полной компиляции.

### ***Нисходящая версия процесса восходящей компиляции***

С помощью нисходящей компиляции один разработчик компилирует в программе весь проект. Вы можете использовать нисходящий процесс для одновременной оптимизации всех блоков проекта, или для оптимизации одного или нескольких критических блоков проекта или IP ядер прежде, чем добавить остальной проект. Вы можете зафиксировать результаты компоновки и характеристики завершённых блоков, пока редактируются другие части проекта, это также сократит время каждой итерации. Различные разработчики или поставщики IP могут создавать и верифицировать свой HDL код независимо, но только один человек (руководитель проекта или системный администратор) компилирует и оптимизирует его в качестве проекта Quartus II верхнего уровня.

С помощью нисходящего проектирования, индивидуальные разработчики или поставщики IP могут закончить разработку и оптимизацию своих проектов в виде отдельных Quartus II проектов, а затем интегрировать каждый низкоуровневый проект в единственный головной проект. Инкрементная компиляция содержит средства импорта и экспорта, необходимые для использования этого метода. Разработчики низкоуровневых блоков могут экспортировать оптимизированные по размещению и разводке списки соединений для своего проекта, вместе с назначениями, именуемыми LogicLock регионами. Руководитель проекта импортирует каждый блок проекта в головной проект в качестве раздела проекта.

Процесс восходящего проектирования позволяет использовать результаты компиляции других проектов, с целью сохранения рабочих характеристик и уменьшения времени компиляции в многопользовательской среде разработчиков.

Процесс восходящего проектирования содержит также потенциальные изъяны, которые требуется предусмотреть:

- Достижение временных ограничений для целого проекта может быть очень затруднено, если во время компиляции низкоуровневых блоков, вы не использовали информацию о других блоках проекта. Этой проблемы можно избежать, если точно рассчитать временные ограничения и применить специальное управление проектом, такое как, постоянная регистрация портов на границах модуля.

- По этим причинам, планирование ресурсов и размещения позволяет избегать избыточности и конфликтов ресурсов. Создание архитектуры – пожалуй, самая важная операция в восходящем проектировании.

В процессе восходящего проектирования, руководитель головного должен разработать план проекта, а затем передать ограничения для разработчиков низкоуровневых блоков с помощью генерируемых Quartus II скриптов разделов проекта.

Когда используется процесс полной инкрементной компиляции, пользователи, которые традиционно использовали восходящий метод в качестве основной возможности сохранения рабочих характеристик, могут применить нисходящий метод для достижения тех же целей. Эти намного лучше по двум причинам:

- Выполнение нисходящего процесса много проще, чем его восходящий эквивалент. Например, исключается необходимость импорта или экспорта низкоуровневых проектов.
- Метод нисходящего проектирования снабжает программу проекта информацией о каждом подпроекте, таким образом, она может выполнять глобальное размещение и оптимизацию соединений. Что позволяет упростить достижение хороших характеристик в процессе нисходящего проектирования по сравнению с процессом восходящего проектирования.

Средства инкрементной компиляции программы Quartus II достаточно гибкие и поддерживают различные методы проектирования. Вы можете комбинировать восходящее и нисходящее проектирования внутри одного проекта. Если головной проект содержит один или несколько блоков, которые оптимизируются независимыми разработчиками или IP поставщиками, вы можете импортировать эти блоки (используя восходящее проектирование) в проект, который также имеет разделы для нисходящего проектирования. В дополнение, когда вы выполните временные ограничения для проекта, вы можете создать подпроект из одного блока проекта, для того, чтобы другой разработчик оптимизировал его в отдельном проекте Quartus II, и снабдить его информацией об остальном проекте, чтобы в подпроекте добиться лучших характеристик. Следуя комбинированной методике проектирования, вы получаете преимущество в командной производительности при восходящем проектировании, когда используете преимущество нисходящего проектирования для большинства логики проекта.

Восходящая инкрементная компиляция не поддерживается в чипах **HardCopy** или FPGA, если установлены настройки **migration device** (перемещение чипов). Вы не сможете использовать метод восходящего проектирования при перемещении чипов HardCopy ASIC. За подробной информацией обратитесь к "HardCopy компиляция и процесс перемещения" на странице 2-64.

В этом руководстве по быстрому старту описывается процесс нисходящего проектирования, поскольку он более распространен и имеет простой в использовании процесс инкрементной компиляции. За подробной информацией обратитесь к "Экспорт и импорт разделов" на странице 2-29.