

## **Компиляция проекта**

Когда вы добавляете **.stp** файл в ваш проект, встроенный логический анализатор SignalTap II становится частью вашего проекта. Вам нужно скомпилировать ваш проект для объединения с логикой SignalTap II и разрешить соединение JTAG, которое используется для контроля логическим анализатором. Когда вы используете для отладки обычный внешний анализатор, то часто приходится делать изменения для отображения сигналов, а также в состоянии триггеров. Поскольку эти настройки зачастую эти настройки сказываются на времени перекомпиляции, то когда используете встроенный логический анализатор SignalTap II, используйте его вместе с инкрементной компиляцией в программе Quartus II, чтобы уменьшить время перекомпиляции.

### **Быстрые компиляции с инкрементной компиляцией Quartus II**

Для использования инкрементной компиляции совместно со встроенным логическим анализатором SignalTap II, выполните следующие шаги:

1. Разрешите **Полную инкрементную компиляцию** для вашего проекта.
2. Назначьте разделы проекта.
3. Установите разделам подходящие уровни сохранения компоновки.
4. Разрешите **SignalTap** в вашем проекте.
5. Добавьте сигналы в **SignalTap**, используя соответствующий фильтр списков соединений в поиске узлов (либо SignalTap II: пре-синтез, либо SignalTap II: пост-компоновка.)

Когда вы компилируете ваш проект с **.stp** файлом, объекты `sld_signaltap` и `sld_hub` автоматически добавляются в иерархию компиляции. Два этих объекта – это основные компоненты встроенного логического анализатора SignalTap II, создающих логику триггеров и интерфейс JTAG, необходимые для операций.

Инкрементная компиляция позволяет вам сохранять результаты синтеза и компоновки вашего оригинального проекта, и добавлять встроенный логический анализатор SignalTap II в ваш проект без перекомпиляции оригинального исходного кода. Это средство также полезно, когда вы хотите изменить конфигурацию **.stp** файла. Например, вы можете изменить глубину буфера отсчётов или тип памяти без выполнения полной компиляции, после того как сделаете изменения. Только встроенный логический анализатор SignalTap II, сконфигурированный как отдельный раздел, должен быть перекомпилирован, чтобы отразить изменения.

Для использования инкрементной компиляции, сначала разрешите **Полную инкрементную компиляцию** в вашем проекте, если она ещё не разрешена, назначьте разделы проектов, если нужно, и установите разделам проектов корректный уровень сохранения. Инкрементная компиляция – это настройка по умолчанию для новых проектов в программе Quartus II, так что вы создаёте разделы прямо в новом проекте. Однако, совсем не требуется создавать несколько разделов проекта, чтобы использовать средство инкрементной компиляции SignalTap II. Когда в вашем проекте установлено использовать полную инкрементную компиляцию, встроенный логический анализатор SignalTap II функционирует как отдельный раздел проекта. Вы можете выгодно использовать инкрементную компиляцию, если используете **фильтр SignalTap II: пост-компоновка** в поиске узлов для добавления сигналов к логическому анализу.

### Разрешение инкрементной компиляции в вашем проекте

Чтобы разрешить инкрементную компиляцию, если она ещё не разрешена, выполните следующие шаги:

1. В меню Назначения, кликните на окно **Разделы проекта**.
2. В списке **Инкрементная компиляция**, выберите **Полная инкрементная компиляция**.
3. Создайте пользовательские разделы, если нужно, и установите тип списка соединений **пост-компоновка** для всех разделов.

Тип списка соединений для головного раздела по умолчанию установлен как **исходный**. Чтобы использовать преимущества инкрементной компиляции, установите тип списка соединений от которых вы собираетесь делать отводы, как **пост-компоновка**.

4. В меню Процессы, кликните **Старт компиляции**, или на панели инструментов кликните **Старт компиляции**.

Ваш проект сначала полностью скомпилируется, появятся созданные вами разделы проекта. Если разрешён встроенный логический анализатор SignalTap II для вашего проекта, то он всегда будет в отдельном разделе. После первой компиляции, вы сможете использовать встроенный логический анализатор SignalTap II для анализа сигналов из списка соединений пост-компоновки. Если ваши разделы настроены верно, последующие компиляции с настройками SignalTap II займут меньше времени.

За дополнительной информацией о конфигурировании и выполнении инкрементной компиляции, обратитесь к главе «Инкрементная компиляция Quartus II для иерархических и командных проектов» в томе 1 Настольной книги Quartus II.

## Использование инкрементной компиляции со встроенным логическим анализатором SignalTap II

Встроенный логический анализатор SignalTap II автоматически сконфигурирован для работы с процессом инкрементной компиляции. Для всех сигналов, которые вы хотите подключить к встроенному логическому анализатору SignalTap II из списка соединений пост-компоновка, установите тип списка соединений для разделов, содержащих нужные сигналы **пост-компоновка** или **пост-компоновка (строго)** с уровнем сохранения разводки **размещение и разводка** в Окне разделов проекта. Используйте **фильтр SignalTap II: пост-компоновка** в **Поиске узлов** для добавления интересующих вас сигналов в файл конфигурации SignalTap II. Если вы хотите добавить сигналы из списка соединений пре-синтез, установите тип списка соединений **исходный файл** и используйте **фильтр SignalTap II: пре-синтез** в **Поиске узлов**. Не используйте тип списка соединений **пост-синтез** для встроенного логического анализатора SignalTap II.

Подчиняйтесь правилам следующих руководств, когда используете узлы пост-компоновки и пре-синтеза:

- Прочитайте все руководства по инкрементной компиляции, чтобы создать правильный раздел для проекта.
- Для ускорения времени компиляции, используйте только узлы пост-компоновки для разделов с установленным уровнем сохранения пост-компоновка.
- Не смешивайте узлы пре-синтез и пост-компоновка в разделе. Если вам нужен отвод от узлов пре-синтез для одного раздела, сделайте все отводы от узлов в этом разделе типа пре-синтез и измените тип списка соединений для раздела на **исходный** в окне разделов проекта.

Имена узлов могут различаться в списках соединений пост-синтез и пост-компоновка. Вообще, сигналы регистров и пользовательских выводов разделяют имена между двумя списками соединений. Во время компиляции, оптимизация изменяет имена комбинационных сигналов в вашем RTL. Если тип выбранного имени узла отсутствует в типе списка соединений, компилятор не сможет найти сигнал для подключения к элементу встроенного логического анализатора SignalTap II для анализа. Компилятор выдаст критическое предупреждение, чтобы защитить вас от этого сценария. Сигнал, который не подключен, соединяется с землёй на вкладке **данных SignalTap II**.

Если вы не используете процесс инкрементной компиляции со встроенным логическим анализатором SignalTap II, и не требуется изменять исходные файлы, знайте, что вы можете удалить сгенерированные компилятором имена узлов пост-компоновки. Изменения исходного кода оказывают влияние на раздел через повторный синтез. Во время синтеза, компилятор может не найти сгенерированные компилятором имена узлов от предыдущей компиляции.

Altera рекомендует использовать только зарегистрированные и пользовательские входные сигналы в качестве отладочных отводов в вашем **.stp** файле, насколько это возможно.

Регистрированные и пользовательские входные сигналы используют общие имена узлов в списках соединений пост-компоновка и пре-синтез. В результате, использование только зарегистрированных и отведённых пользовательских входных сигналов в вашем **.stp** файле ограничит изменения, которые вам необходимо сделать в вашей конфигурации SignalTap II.

Вы можете проверить узлы, которые подключены к каждому элементу SignalTap II, используя отчёт компиляции отладки в системе. В этом отчёте - список выбранных вами узлов, подключенных к элементу SignalTap II, тип используемого списка соединений для правильного подключения, и текущее имя узла, используемое после компиляции. Если инкрементная компиляция выключена, отчёты отладки в системе размещаются в папке Анализ и Синтез. Если инкрементная компиляция включена, эти отчёты размещаются в папке Объединение разделов. На рисунке 14-42 показан пример отчета компиляции отладки в системе для проекта, использующего инкрементную компиляцию.

**Figure 14-42.** Compilation Report Showing Connectivity to SignalTap II Instance

Name	Type	Status	Partition Name	Netlist Type Used	Actual Conne...	Details	
1	sr[62][5]	pre-synthesis	connected	Top	post-fit	sr[62][5]	N/A
2	sr[62][5]	pre-synthesis	connected	Top	post-fit	sr[62][5]	N/A
3	sr_tap_one[0]	pre-synthesis	connected	Top	post-fit	altshift_t...	N/A
4	sr_tap_one[0]	pre-synthesis	connected	Top	post-fit	altshift_t...	N/A
5	clk	post-fitting	connected	Top	post-fit	clk~input	N/A
6	sr[0][0]	post-fitting	connected	Top	post-fit	sr[0][0]	N/A
7	sr[0][0]	post-fitting	connected	Top	post-fit	sr[0][0]	N/A
8	sr[1][0]	post-fitting	connected	Top	post-fit	sr[1][0]	N/A
9	sr[1][0]	post-fitting	connected	Top	post-fit	sr[1][0]	N/A
10	sr[1][1]	post-fitting	connected	Top	post-fit	sr[1][1]	N/A
11	sr[1][1]	post-fitting	connected	Top	post-fit	sr[1][1]	N/A
12	sr[1][2]	post-fitting	connected	Top	post-fit	sr[1][2]	N/A
13	sr[1][2]	post-fitting	connected	Top	post-fit	sr[1][2]	N/A
14	sr[1][3]	post-fitting	connected	Top	post-fit	sr[1][3]	N/A
15	sr[1][3]	post-fitting	connected	Top	post-fit	sr[1][3]	N/A

Чтобы удостовериться, что ваш оригинальный проект не изменён, изучите сообщения в секции **объединение разделов** отчёта компиляции. На рисунке 14-43 показан пример отображаемого сообщения.

**Figure 14-43.** Compilation Report Messages

Messages: System (16) Processing (178) Extra Info \ Info (155) \ Warning (19) \ Critical Warning (4) \ Error \ Suppressed (6) \ Flag /

Message: 203 of 412

Partition Name	Compilation Hierarchy Path	Netlist Type	Filter Preservation Level	Color	Post-Synthesis Netlist Status	Post-Fit Netlist Stat
Design Partitions						
<<new>>						
Top	filtrf	Pcst-Fit	Placement and Routing		14:08:37 Aug 08, 2007	10:32:57 Aug 09, 2007

Пока вы не сделаете изменения в раздел проекта, которому потребуется перекомпиляция, будет перекомпилироваться только раздел проекта SignalTap II. Если вы делаете дальнейшие изменения только в **.stp** файле, должен перекомпилироваться только раздел проекта SignalTap II, уменьшая ваше время на перекомпиляцию.

### **Предотвращение изменений вынужденной перекомпиляции**

Вы можете сконфигурировать **.stp** файл так, чтобы защитить его от изменений, которые обычно побуждают перекомпиляцию. Чтобы сделать это, выберите режим *бронирования* рядом с списком узлов на вкладке **Установки**. Для бронирования вашей конфигурации, выберите *позволять только изменения состояния триггера*, не зависимо от того, будете ли вы использовать инкрементную компиляцию.

За подробной информацией об использовании режима бронирования, обратитесь к помощи Quartus II.

### **Сохранение временных параметров во встроенном логическом анализаторе SignalTap II**

В дополнении к функциональной верификации, временные ограничения – это один из очень критичных процессов в удачно реализованном проекте. Когда вы компилируете проект со встроенным логическим анализатором SignalTap II без использования инкрементной компиляции, вы добавляете IP в существующий проект. Поэтому, вы можете повлиять на существующее расположение, разводку и временные характеристики в вашем проекте. Чтобы минимизировать влияние, которое оказывает встроенный логический анализатор SignalTap II на ваш проект, Altera рекомендует вам использовать для проекта инкрементную компиляцию. Инкрементная компиляция – это настройка по умолчанию для новых проектов, она легко добавляется и настраивается в существующих проектах. Встроенный логический анализатор SignalTap II, находясь в своём разделе, не оказывает и малейшего влияния на ваш проект.

В дополнении к использованию процесса инкрементной компиляции в вашем проекте, вы можете воспользоваться следующей техникой для достижения временных характеристик:

- Избегайте добавление критических путей сигналов в ваш **.stp** файл.
- Минимизируйте количество комбинационных сигналов, добавленных в ваш **.stp** файл, и старайтесь добавлять регистры.
- Определите ограничение Fmax для всех тактов в вашем проекте.

Примеры сохранения временных характеристик со встроенным логическим анализатором SignalTap II, находятся в главе «Оптимизация площади и времени» в томе 2 Настольной книги Quartus II.

### **Анализ ресурсов и характеристик**

Есть обязательный компромисс между гибкостью встроенного логического анализатора SignalTap II в запущенном времени, его временными характеристиками и использовании ресурсов. Встроенный логический анализатор SignalTap II позволяет вам выбрать параметры, конфигурируемые в запущенном времени, для балансирования между необходимой гибкостью в запущенном времени, скоростью и площадью. Значения, заданные по умолчанию, выбирают предоставление максимальной гибкости, чтобы вы могли завершить отладку по возможности быстро; однако, вы можете отрегулировать эти настройки, чтобы в дальнейшем иметь более оптимальную конфигурацию для вашего проекта.

Советы в этой главе содержат некоторые указания, дающие специальный временной запас, если вы определили логике SignalTap II критический путь, или уменьшают использование ресурсов встроенным логическим анализатором SignalTap II, если ваш проект стеснён в ресурсах.

Если логика SignalTap II является частью критического пути, следующие советы помогут вам ускорить характеристики встроенного логического анализатора SignalTap II:

■ **Запретите опции конфигурирования в запущенном времени** - определённые ресурсы размещаются для предоставления гибкости в запущенном времени. Если вы используете процессы расширенных триггеров или триггеров базовых состояний, запретите параметры конфигурирования в запущенном времени для повышения Fmax логики SignalTap II. Если вы используете процесс базовых состояний триггера, попробуйте запретить опцию **Перейти в состояние назначения** и выполните перекомпиляцию, прежде чем запрещать остальные опции конфигурирования в запущенном времени. Опция **Перейти в состояние назначения** сильнее влияет на Fmax, чем другие опции конфигурирования в запущенном времени.

■ **Минимизируйте количество сигналов, которые имеют выбор разрешения триггера** – все сигналы, которые вы добавляете в .stp файл имеют включенную опцию разрешения триггера. Выключите разрешение для триггера для тех сигналов, которые вы не намерены использовать в качестве триггеров.

■ **Включите физический синтез для восстановления синхронизации регистров** – Если у вас разрешено большое количество сигналов триггера (больше, чем количество входов, которые могут быть размещены в LAB), которые разветвляются по входу на логику состояний базовых вентилей триггера, таких как базовые состояния триггера или оператор логического уменьшения во вкладке расширенного триггера, включите опцию **Выполнить восстановление синхронизации регистров**. Это может помочь сбалансировать комбинационную логику между LABs.

Если ваш проект стеснён в ресурсах, следующие рекомендации могут помочь вам освободить логику или память, используемую логическим анализатором SignalTap II:

■ **Запретите опции конфигурирования в запущенном времени** - Запрет конфигурирование в запущенном времени для расширенных состояний триггера или опции конфигурирование в запущенном времени для процесса базовых состояний триггера выделит из использования несколько Les.

■ **Минимизируйте количество сегментов в буфере захвата** – Вы можете уменьшить количество логических ресурсов, используемых логическим анализатором SignalTap II, ограничив количество сегментов в вашем буфере захвата, оставив только нужное.

■ **Запретите разрешение данных для сигналов, которые используются только как триггеры** – По умолчанию, обе опции **разрешение данных** и **разрешение триггера** выбраны для всех сигналов. Выключая опцию разрешения данных для сигналов, используемых только как входы триггера, вы освобождаете ресурсы памяти, используемые логическим анализатором SignalTap II.

Поскольку характеристики зависят от проекта, попробуйте эти опции в различных комбинациях, пока не добьётесь нужного баланса между функциональностью, характеристикам и использованием ресурсов.

За дополнительной информацией об оптимизации площади и временных характеристик, обратитесь к главе «Оптимизация площади и времени» в томе 2 Настольной книги Quartus II.