

## **Генерация скриптов раздела восходящего проектирования для менеджера проекта**

Скрипты разделов в восходящем проектировании автоматизируют процесс переноса информации головного проекта низкоуровневым модулям. Программа Quartus II снабжает этим интерфейсом для управления ресурсами и временным планированием в проекте верхнего уровня. Это упрощает разработчикам низкоуровневых проектов применение инструкций руководителя проекта, и позволяет избежать конфликтов при импорте и объединении проектов в головном проекте. Этим снижается необходимость дальнейшей оптимизации проекта после интеграции и значительно увеличивается продуктивность совместной работы разработчиков.

Генерация скриптов разделов в восходящем проектировании является дополнительной опцией.

Примеры некоторых сценариев проектирования, использующих скрипты, в главе "Выполнение командного проекта в процессе восходящего проектирования" на странице 2-53. Во время типового процесса разработки, руководитель проекта должен выполнить некоторые или все рекомендации, чтобы выполнить успешную интеграцию подпроектов:

- Установить вручную, какими назначениями из головного проекта должны быть снабжены низкоуровневые проекты. Здесь необходимы точные знания о том, какие назначения программа Quartus II требует установить для низкоуровневых проектов.
- Вручную снабдить низкоуровневый проект назначениями из головного проекта. Это требует детального знания Tcl или других языков скрипирования для эффективного снабжения ограничениями.
- Определить вручную необходимые временные и локальные назначения, которые помогут обойти условные ограничения восходящего проектирования. Это потребует изучение логики низкоуровневого проекта для определения необходимых временных ограничений.
- Выполнить финальное временное закрытие<sup>1</sup> и избежать конфликтов размещения в головном проекте. В виду того, что низкоуровневые проекты не имеют информацию друг о друге, достижение требуемых ограничений проектами нижнего уровня не гарантирует того, что они будут достигнуты при интеграции в головном проекте. Такой результат может получить руководитель проекта, который не обладал информацией о размещении низкоуровневых проектов.

Использование программы Quartus II для генерации скриптов проекта восходящего проектирования для головного проекта, делает эту задачу проще и исключает шансы на ошибки во время сообщения между руководителем проекта и низкоуровневыми разработчиками. Скрипты разделов передают назначения, сделанные в головном проекте, и создают некоторые новые ограничения размещения, помогая разработчикам видеть, как их проект соединяется с другими разделами. Если потребуется, вы можете исключить некоторые разделы проекта.

---

<sup>1</sup> Временное закрытие (timing closure) – это процесс FPGA проектирования, при котором путём изменения архитектуры размещения проекта внутри чипа, достигаются требуемые временные характеристики работы схемы.

Генерируйте скрипты проекта после выполнения успешной компиляции модуля головного проекта. В меню **Проект** выберите **Генерировать скрипты раздела восходящего проектирования**. Проект может содержать пустые разделы, в качестве резервации места под низкоуровневые разделы, поэтому вы можете выполнить **Начальную временную оценку**, являющуюся частью полной компиляции, для экономии времени компиляции.

Следующие подразделы содержат информацию, которая может быть включена в скрипты разделов Tcl восходящего проектирования. Используйте опции во вкладке **Генерировать скрипты раздела восходящего проектирования** для выбора типов назначений, которые вы оставите и воссоздадите в разделах низкоуровневых проектов. В любое время вы можете перезапустить процесс генерации скриптов, программа Quartus II пересоздаст файлы и заменит старые версии этих файлов.

За подробной информацией об текущих условных ограничениях в скриптах проекта восходящего проектирования, обратитесь к главе "Ограничения в скриптах восходящего проектирования" на странице 2-67.

## Создание проекта

Вы можете использовать опцию **создать низкоуровневый проект, если он отсутствует** для использования скрипта проекта при создании низкоуровневого проекта, когда требуется. Файл Проекта Quartus II для каждого низкоуровневого проекта имеет то же имя, что и соответствующий раздел проекта.

Это средство создания проекта, скрипты работают сами для создания нового проекта, или являются исходными для создания назначений в существующем проекте.

## Исключенные разделы

Используйте опцию **Исключённые разделы** сверху диалогового окна для исключения отдельных разделов из процесса генерации скрипта Tcl. Используйте кнопку браузера, затем выделите имя раздела в диалоговом окне **Выбор Разделов**, а потом используйте соответствующие кнопки для выбора (отмены выбора) необходимых разделов.

## Назначения из головного проекта

По умолчанию, любые назначения, сделанные в головном проекте (не включая назначения по умолчанию или назначения информации о проекте) сохраняются в скриптах для соответствующих проектов нижнего уровня. Программа использует назначения переменных и определяет логические разделы, которым принадлежат назначения. Это включает в себя: глобальные назначения, назначения для блоков и назначения на уровне элементов. Затем программа изменяет назначения, чтобы проект с его логикой разделов был синтаксически правильно интерпретирован в качестве головного модуля.

Имена файлов проекта, которые относятся к определённому разделу, добавляются к каждому проекту нижнего уровня.

Скрипт использует имена файлов, определённых в головном проекте. Если головной проект использует заголовочный файл для резервирования площади, с именем, отличающимся от имени низкоуровневого проекта, то необходимо добавить соответствующий файл в низкоуровневый проект.

Процесс скрипирования корректно использует назначения для "дикой карты", позволяя использовать только одну "дикую карту". Назначения более чем одной "дикой карте" игнорируются, о чём выводится предостерегающее сообщение.

Используйте следующие опции для определения, какие типы назначений сохраняются для проектов нижнего уровня:

■ **Временные назначения** – если опция включена, все глобальные временные назначения Классического Временного Анализатора для низкоуровневого проекта включены в скрипт, включая ограничения  $t_{co}$ ,  $t_{su}$  и  $f_{MAX}$ . К тому же, после того, как вы скомпилировали проект, используя ограничения **TimeQuest**, генерируются отдельные скрипты создания **.sdc** файла для каждого проекта нижнего уровня, которые снабжают его ограничениями по тактовому сигналу и некоторым максимальным и минимальным задержкам. Это опция может содержать в себе также временные ограничения соединений с внутренними разделами.

■ **Назначения для разделов проекта** – если опция включена, назначения для скрипта, связанного с разделом в низкоуровневом проекте включают в себя, например, назначения для регионов LogicLock.

■ **Назначения локализации выводов** – если опция включена, все назначения локализации для портов в низкоуровневом проекте, которые подключены к выводам чипа в головном проекте, включены в скрипт, таким образом, контролируется перерасход I/O выводов в головном проекте во время фазы интеграции и сохранения размещения.

### Назначения виртуальных выводов

Если включена опция **Создавать виртуальные выводы для низкоуровневых портов, подключенных к другим элементам проекта**, программа Quartus II отыскивает список соединений раздела, и идентифицирует все порты, имеющие межраздельную зависимость. Каждому выводу низкоуровневого раздела ассоциируется внутренний порт другого раздела или головного проекта, скрипт генерирует назначения виртуального вывода, обеспечивая точное расположение, поскольку виртуальные выводы напрямую не подключены к I/O портам в головном проекте. Эти выводы убираются из списка соединений проекта нижнего уровня, когда он импортируется в головной проект.

#### Временные и локальные назначения для виртуальных выводов

Одним из главных проблем метода восходящего проектирования является то, что каждый индивидуальный блок проекта не имеет информации о том, как он подключен к другим блокам проекта. Если вы включили опцию записи назначений виртуальных выводов, то вы также включили опцию ограничения этих виртуальных выводов, для достижения лучших временных характеристик, после того как низкоуровневый проект будет интегрирован в головной проект.

Когда включена опция **Размещать виртуальные выводы с положением исходника/стока в верхнем уровне**, создается скрипт, включающий информацию об ограничении локализации для каждого виртуального вывода. Виртуальные выходы назначены локализации выводам назначений головного проекта, а виртуальные входы назначены локализации исходным выводам головного проекта. Если головной проект использует **Пустые** разделы, окончательное положение соединений неизвестно, но на выводах остаются назначения регионов LogicLock, которые определены как исходные выводы или выводы назначений.

В результате, те виртуальные выводы, которые не могут быть размещены внутри LogicLock региона проекта нижнего уровня, но имеют размещение в головном проекте, исключают затраты ресурсов в низкоуровневом проекте и дают больше информации о низкоуровневом проекте и зависимостях его портов. Это размещение ограничений не может быть импортировано в головной проект.

Когда включены **Добавить максимальную задержку созданным виртуальным входам, Добавить максимальную задержку от созданных виртуальных выходов**, или и то и другое, в скрипт включены временные ограничения для каждого созданного виртуального вывода. Величина, которую вы вводите в диалоговом окне, - это максимальная задержка к или от всех путей между виртуальными выводами, чтобы помочь добиться требуемых временных ограничений для законченного проекта. Программа использует назначения INPUT\_MAX\_DELAY или OUTPUT\_MAX\_DELAY для применения ограничений.

Эта опция позволяет руководителю проекта рассчитать основной временной бюджет для всех внутренних соединений между выводами проектов нижнего уровня. Разработчик проекта нижнего уровня может игнорировать эти ограничения и применить назначения для некоторых выводов на уровне узлов проекта, когда это требуется.

### Назначения регионов LogicLock

Когда включена опция **Копировать назначения регионов LogicLock из верхнего уровня**, скрипт содержит назначения идентифицирующие назначение LogicLock для раздела.

Скрипт также сохраняет назначения для создания LogicLock регионов для всех других разделов. Если включена опция **Включить все регионы LogicLock в низкоуровневых проектах**, скрипт для каждого раздела включает в себя назначения всех регионов LogicLock для головного проекта и для каждого низкоуровневого раздела, воспроизводящий архитектуру законченного проекта в каждом разделе. Регионы, не принадлежащие другим разделам, содержат виртуальные выводы, отображающие исходные порты и порты назначений междраздельных соединений. Это позволяет каждому разработчику видеть соединения между его разделом и другими разделами в головном проекте, и помогает найти и минимизировать влияние источника конфликтов в верхнем уровне.

Если включена опция **Убрать имеющиеся регионы LogicLock из низкоуровневых проектов**, скрипт содержит команду, которая удаляет регионы LogicLock из низкоуровневых разделов перед запуском скрипта. Это обеспечивает условие, по которому, регионы LogicLock, которые не являются частью головного проекта, не будут частью конечного проекта, что помогает избежать конфликтов, когда низкоуровневые разработчики используют регионы LogicLock, определенные в головном проекте.

### Назначения продвижения глобальных сигналов

Для предотвращения конфликтов при использовании глобальных сигналов во время импорта в головной проект, вы можете выбрать запись назначений, контролирующих, какие сигналы продвигаются в глобальные ресурсы трассировки в низкоуровневых разделах. Эта опция помогает спланировать расход глобальных ресурсов трассировки.

Если включена опция **Продвигать глобальные сигналы из головного в низкоуровневые проекты**, программа Quartus II отыскивает список соединений раздела и идентифицирует глобальные ресурсы, включая тактовые сигналы. Для соответствующих разделов, скрипт, содержащий назначения продвижения глобальных сигналов, снабжает проект нижнего уровня информацией о размещении глобальных ресурсов.

Когда включена опция **Запрет автоматического глобального продвижения в низкоуровневые проекты**, скрипт содержит назначения, которые выключают все настройки глобального продвижения в низкоуровневые проекты. Эти настройки содержат логическую опцию **Автоматические контрольные сигналы глобальной памяти**, опцию логического разрешения выходов и контроля над продвижением тактовых и регистровых сигналов. Если вы выбрали опцию **Запрет автоматического глобального продвижения в низкоуровневые проекты** совместно с **Продвигать глобальные сигналы из головного в низкоуровневые проекты**, вы сможете определить, какие именно сигналы из тех, что продвигаются к глобальным ресурсам в головном проекте, продвигаются также и в низкоуровневых проектах.

### Генерация сборочного файла

Сборочные файлы позволяют вам использовать команду **Делать** для определения, какой проект в восходящем проектировании является новейшим, если вы установили соответствующую утилиту на свой компьютер. Опция **Генерация сборочного файла для обслуживания низкоуровневых и высокоуровневых проектов** создает сборочный файл для каждого раздела в головном проекте, а также ведущий сборочный файл, который может запускать сборочные файлы низкоуровневых проектов. Программа Quartus II располагает ведущий сборочный файл в корне директории проекта, а сборочные файлы разделов в соответствующих директориях низкоуровневых проектов.

Вы можете определить зависимости в сборочных файлах, показывающие, какой исходный файл должен быть ассоциирован с каким разделом. Сборочные файлы используют размещение в директории сгенерированное с использованием опции **Создать низкоуровневый проект, если он ещё не существует**. Если вы создаете свои низкоуровневые проекты без использования этой опции, вам нужно модифицировать переменные в заголовке сборочного файла для определения размещения директории каждого низкоуровневого проекта.

Для запуска сборочных файлов, используйте команду `make -f master_makefile.mak` в выходной директории скриптов. Ведущий сборочный файл сначала запускает отдельные сборочные файлы для каждого раздела, исходными для которых являются Tcl скрипты, а потом генерирует файл **.qxr** для экспорта проекта в виде раздела. Далее запускается сборочный файл верхнего уровня, который определяет эти новые **.qxr** файлы как файлы импорта для соответствующих разделов в головном проекте. Затем сборочный файл верхнего уровня импортирует результаты нижнего уровня и выполняет полную компиляцию, создающую конечный проект.

Для исключения некоторых разделов из процесса компиляции, отредактируйте секцию `EXCLUDE_FLAGS` в **master\_makefile.mak** согласно инструкциям в файле и определите необходимые свойства. Вы можете также исключить эти разделы из процесса компоновки, экспорта или импорта с использованием команды **Делать**. Для исключения раздела, запустите сборочный файл, используя команду, называемую утилитой GNU make, как показано в примере:

```
gnumake -f master_makefile.mak exclude_<partition directory>=1 ←
```

Эта команда содержит инструкцию, позволяющую не создавать разделы из имеющихся файлов в `<partition directory>`. Несколько директорий могут быть исключены добавлением команды `multiple exclude_<partition directory>`. Опции командной строки подменяют другие опции сборочного файла.

Другим свойством сборочного файла является возможность иметь ведущий сборочный файл, вызывающий низкоуровневые сборочные файлы параллельно в системах с несколькими процессорами. Эта опция позволяет значительно сократить время компиляции разработчикам, работающим на мультипроцессорных ПК. В утилите **GNU make**, добавьте флаг `-j<N>` в команду **Делать**. Значение `<N>` - это число процессоров, которое вы собираетесь загрузить работой.,

Сборочные файлы не содержат опцию очистки команды **Делать**, поэтому проект может быть перекомпилирован, когда команда **Делать** запущена снова, а файл **.qxr** уже существует.