

## Поддержка скриптов

Запускайте процедуры и делайте настройки, используя Tcl скрипт, как это описано в этой главе. Вы можете также запускать эти процедуры из командной строки. Подробная информация об опциях команд скриптирования, обратитесь к браузеру помощи командной строки Quartus II и Tcl API. Для запуска браузера помощи, наберите следующую команду в командной строке:

```
quartus_sh --qhelp ↵
```

Руководство ссылок скриптирования содержит эту же информацию в формате PDF.

Больше информации о Tcl скриптировании, находится в главе "Tcl скриптирование" томе 2 настольной книги Quartus II. Больше информации обо всех настройках и ограничениях в программе Quartus II, обратитесь к Ссылке на руководство файла настроек Quartus II. Больше информации о скриптировании в командной строке, обратитесь к главе "Скриптирование в командной строке" в томе 2 настольной книги Quartus II.

## Создание вывода SignalProbe

Для создания вывода SignalProbe, наберите следующую команду:

```
make_sp [-h | -help] [-long_help] [-clk <clk>] [-io_std <io_std>] \
-loc <loc> -pin_name <pin name> [-regs <regs>] [-reset <reset>] \
-src_name <source name> ↵
```

## Удаление вывода SignalProbe

Для удаления вывода SignalProbe, наберите следующую команду:

```
delete_sp [-h | -help] [-long_help] -pin_name <pin name> ↵
```

## Разрешение вывода SignalProbe

Для разрешения вывода SignalProbe, наберите следующую команду:

```
enable_sp [-h | -help] [-long_help] -pin_name <pin name> ↵
```

## Запрещение вывода SignalProbe

Для запрещения вывода SignalProbe, наберите следующую команду:

```
disable_sp [-h | -help] [-long_help] -pin_name <pin name> ↵
```

## Выполнение SignalProbe компиляции

Для выполнения SignalProbe компиляции, наберите следующую команду:

```
check_netlist_and_save ↵
```

### Перемещение предыдущих выводов SignalProbe в версии программы Quartus II 6.0 и старше

Для перемещения ранее созданных выводов SignalProbe в версии программы Quartus II 6.0 и старше, наберите следующую команду:

```
convert_signal_probes ↵
```

#### Пример скрипта

В примере 13-3 показан скрипт, который создаёт SignalProbe вывод, названный sp1, и подключает вывод sp1 к исходному узлу reg1 в уже скомпилированном проекте.

#### **Example 13-3. Creating a SignalProbe Pin Called sp1**

```
Package require ::quartus::chip_editor
Project_open project
Read_netlist
Make_sp -pin_name sp1 -src_name reg1
Check_netlist_and_save
Project_close
```

### Использование SignalProbe для семейства чипов APEX™

Семейства чипов APEX не поддерживают создание ECO изменений в списке соединений пост-компоновка. Компиляция SignalProbe может инкрементно разводить внутренние сигналы на выходные выводы. Средство внутренней инкрементной разводки SignalProbe не оказывает влияния на свойства проекта.

Для использования средства SignalProbe, выполните следующие шаги:

1. Зарезервируйте выводы SignalProbe, за подробной информацией обратитесь к главе "Резервирование выводов SignalProbe" на странице 13-2.
2. Назначьте исходник SignalProbe для каждого вывода SignalProbe.
3. Выполните компиляцию SignalProbe.
4. Проанализируйте результаты SignalProbe компиляции.

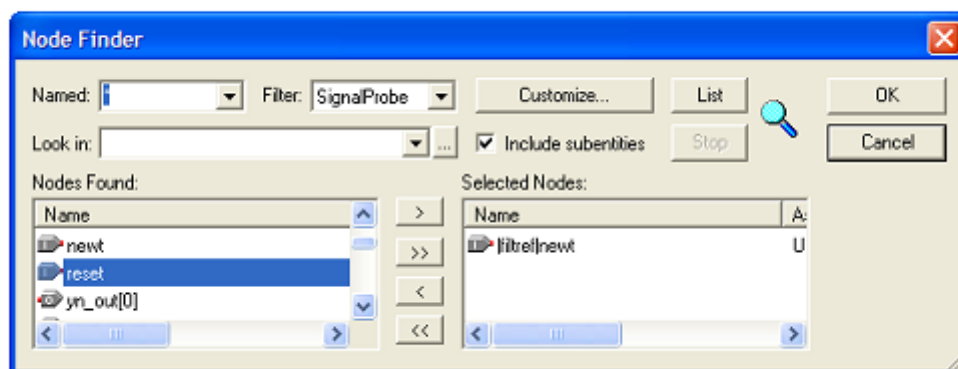
#### Добавление исходников SignalProbe

Исходники SignalProbe – это сигналы в базе данных пост-компиляции проекта с возможностью их разводки на выходной вывод. Для назначения исходника SignalProbe для SignalProbe вывода или для неиспользуемого выходного вывода, выполните следующие шаги:

1. В меню **Инструменты**, кликните на **Выводы SignalProbe**. Откроется диалоговое окно **SignalProbe Выводы**.
2. В списке **Текущие и потенциальные выводы SignalProbe**, выберите вывод SignalProbe, которому вы хотите добавить исходник SignalProbe.
3. Кликните **Просмотр** и выберите исходник SignalProbe.
4. Кликните **ОК**.
5. В диалоговом окне **Выводы SignalProbe**, если исходник ещё не назначен выводу SignalProbe, кликните **Добавить**. Если вывод SignalProbe уже назначен, кликните **Изменить**.
6. Кликните **ОК**.

Диалоговое окно **Поиск узлов** показано с выбранным фильтром SignalProbe (рисунок 13-7). Кликните **Список**, чтобы посмотреть все возможные исходники SignalProbe. Если вы не можете найти определённый узел с помощью фильтра SignalProbe, этот узел был удалён программой Quartus II во время оптимизации или размещён в чипе таким образом, что его не возможно развести до вывода.

**Figure 13–7. Available SignalProbe Sources in the Node Finder**



Когда источник вывода SignalProbe добавлен или изменён, вывод SignalProbe автоматически разрешён. Для запрещения вывода SignalProbe, выключите **SignalProbe разрешены**.

### Выполнение SignalProbe компиляции

После полной компиляции, вы можете вручную запустить SignalProbe компиляцию. Компиляция SignalProbe выполняет следующие функции:

- Подтверждает правильность выводов SignalProbe
- Подтверждает правильность определённых вами исходников SignalProbe
- Если возможно, добавляет регистры в пути SignalProbe
- Пытается развести от исходников SignalProbe, через регистры к выводам SignalProbe

Для автоматического запуска SignalProbe компиляции после полной компиляции, в меню **Инструменты**, кликните на **Выводы SignalProbe**. В диалоговом окне Выводы SignalProbe, включите **Автоматическая разводка SignalProbe сигналов во время компиляции**.

Для ручного запуска SignalProbe компиляции после полной компиляции, в меню **Процессы**, выберите **Старт** и кликните **Старт SignalProbe компиляции**.

Вам необходимо запустить Компоновщик перед выполнением SignalProbe компиляции. Компоновщик генерирует список всех внутренних узлов, которые могут быть использованы в качестве SignalProbe исходников.

Для разрешения или запрещения любого SignalProbe вывода, в диалоговом окне Выводы SignalProbe включите или выключите опцию **Разрешить SignalProbe**.

### Запуск SignalProbe в интеллектуальной компиляции

Запуск интеллектуальной компиляции уменьшает время компиляции за счёт прохождения только необходимых модулей во время компиляции. Поэтому, полная компиляция требуется только, если изменяются некоторые файлы проекта, настройки Анализа и синтеза или Компоновщика.

Для включения интеллектуальной компиляции, в меню **Назначения**, кликните **Настройки**. В списке **Категории**, выберите **Настройки процесса компиляции** и включите **Использовать Интеллектуальную компиляцию**.

Когда вы запускаете SignalProbe компиляцию с включенной опцией интеллектуальной компиляцией, а в файле проекта внесены изменения, или изменения касаются модулей Анализа и Синтеза или Компоновщика, то отображается следующее сообщение:

*Ошибка: Нельзя выполнить SignalProbe компиляцию, поскольку проекту требуется выполнение полной компиляции.*

Вы можете включить опцию интеллектуальной компиляции, которая позволит вам в дальнейшем работать с настройками и файлами проекта.

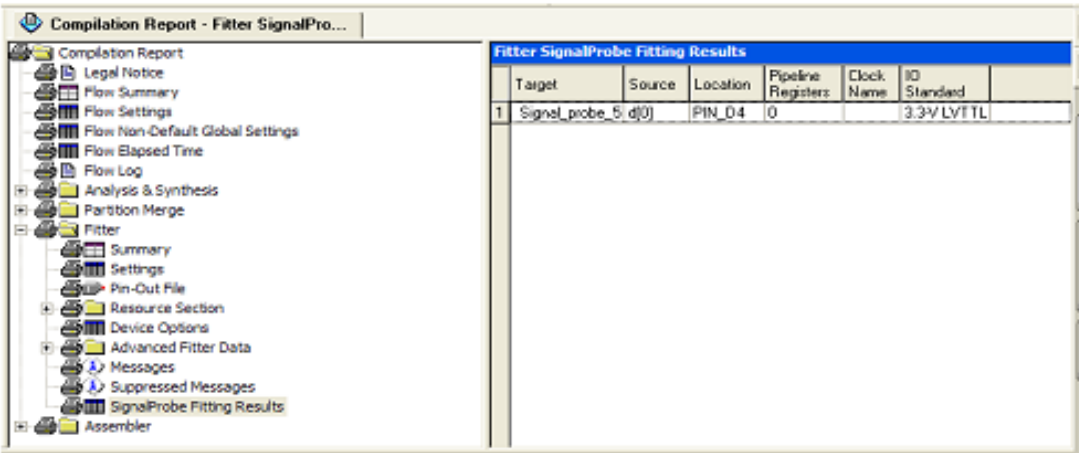
**Осмысление результатов SignalProbe компиляции**

После окончания SignalProbe компиляции, результаты выводятся в двух частях файла отчёта о компиляции. Результаты компоновки и статус каждого вывода SignalProbe (таблица 13-1) показаны на дисплее **Результаты компоновки SignalProbe**, в разделе Компоновщика отчёта о компиляции (рисунок 13-8).

**Таблица 13-1. Значения статуса**

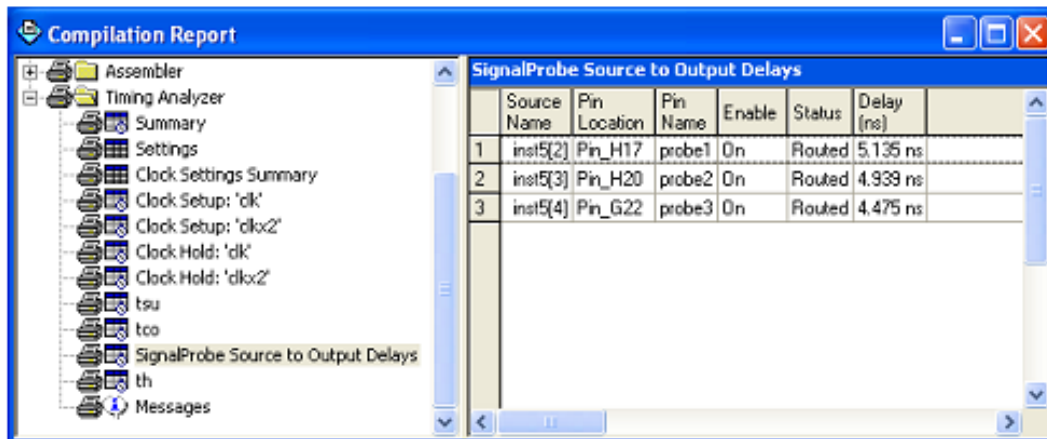
Статус	Описание
Разведён	Подключение и разводка успешны
Не разведён	Не разрешены
Неудачная разводка	Неудачная разводка во время последней SignalProbe компиляции
Требуется компилировать	Изменены назначения после последней SignalProbe компиляции

**Figure 13-8. SignalProbe Fitting Results Page in the Compilation Report Window**



Временные результаты каждого успешно разведенного SignalProbe вывода показаны на дисплее Задержки от источника SignalProbe до выхода в секции Временного анализатора отчёта о компиляции (рисунок 13-9).

**Figure 13–9. SignalProbe Source to Output Delays Page in the Compilation Report Window**



	Source Name	Pin Location	Pin Name	Enable	Status	Delay (ns)	
1	inst5[2]	Pin_H17	probe1	On	Routed	5.135 ns	
2	inst5[3]	Pin_H20	probe2	On	Routed	4.939 ns	
3	inst5[4]	Pin_G22	probe3	On	Routed	4.475 ns	

После SignalProbe компиляции, дисплей процессов окна Сообщения также содержит результаты для каждого SignalProbe вывода, в нём показывается информация о временном резерве для каждого успешно разведённого вывода SignalProbe.

### Средства анализа разводки SignalProbe

Можно начать SignalProbe компиляцию; однако, в силу следующих причин, она не сможет завершиться удачно:

- Разводка недоступна – компиляция SignalProbe безуспешно пытается развести сигнал от источника SignalProbe до вывода SignalProbe, поскольку цепи разводки перегружены
- Неправильный или несуществующий источник SignalProbe – вы ввели источник SignalProbe, который отсутствует или неправильный
- Неподходящий выходной вывод – выбранный выходной вывод признан неподходящим

Средства разводки позволяют определить наличие конфликтов стандартов I/O выводов SignalProbe с другими стандартами I/O в одном банке I/O.

Если перегруженность цепей разводки мешает успешной компиляции SignalProbe, вы можете разрешить компилятору модифицировать разводку для определённых источников SignalProbe. В меню Инструменты, кликните на Выводы SignalProbe и включите Модифицировать последние результаты компоновки во время SignalProbe компиляции. Эта настройка позволит Компоновщику модифицировать существующие каналы разводки, используемые в вашем проекте.

Включённая настройка Модифицировать последние результаты компоновки во время SignalProbe компиляции может изменить характеристики вашего проекта.

### Поддержка скриптов SignalProbe для чипов APEX

Запуск процедур и создание настроек с использованием скрипта Tcl уже описаны в этой главе. Вы можете также запускать эти процедуры из командной строки. Подробная информация об опциях команд скрипирования содержится в браузере Помощи по командной строке Quartus II и Tcl API. Для запуска браузера Помощи, наберите следующую команду в командной строке:

```
quartus_sh --qhelp
```

Руководство ссылок скрипирования содержит эту же информацию в формате PDF.

---

Больше информации о Tcl скрипировании, находится в главе "Tcl скрипирование" томе 2 настольной книги Quartus II. Больше информации обо всех настройках и ограничениях в программе Quartus II, обратитесь к Ссылке на руководство файла настроек Quartus II. Больше информации о скрипировании в командной строке, обратитесь к главе "Скрипирование в командной строке" в томе 2 настольной книги Quartus II.