

Программирование выбранного чипа или чипов

Когда ваш проект, содержащий встроенный логический анализатор SignalTap II, скомпилирован, сконфигурируйте выбранный чип FPGA. Когда вы используете для отладки встроенный логический анализатор SignalTap II, конфигурируйте чип **.stp** файлом, взамен программатора Quartus II. Поскольку вы конфигурируете чип **.stp** файлом, вы не можете открыть более одного **.stp** файла, а программируете несколько чипов для отладки нескольких проектов последовательно.

Настройки в **.stp** файле должны быть совместимы с программным **.sof** файлом, используемым для программирования чипа. Файл **.stp** становится совместимым с программным **.sof** файлом, когда настройки логического анализатора, такие как размер буфера захвата и сигналы, выбранные для мониторинга или защёлкивания, делают возможным программирование выбранного чипа. Если файлы не совместимы, вы всё равно сможете запрограммировать чип, но не сможете запустить или проконтролировать логический анализатор из редактора SignalTap II.

Чтобы гарантировать программную совместимость, убедитесь в программируемости вашего чипа последним **.sof** файлом, сделанным после самой последней компиляции.

Прежде чем начать отладочную сессию, не делайте никаких изменений в настройках **.stp** файла, которые могут потребовать перекомпиляции проекта. Вы можете проверить в статусном окне SignalTap II вверху Менеджера элемента потребует ли изменение проекта перекомпиляцию и создание нового **.sof** файла. Это удобно, можно сделать откат изменений, когда перекомпиляция не нужна. Чтобы защититься от подобных изменений, разрешите режим *бронирования* в **.stp** файле.

Не смотря на то, что проект Quartus II не нужен, он перекомпилируется. База данных проекта содержит информацию об интеграции текущей сессии SignalTap II. Без базы данных проекта, не будет способа проверить, что текущий **.stp** файл создаёт **.sof** файл, который загружается в чип. Если у вас есть **.stp** файл, который не создаёт **.sof** файл, вы увидите некорректные данные захвата во встроенном логическом анализаторе SignalTap II.

Программирование одного чипа

Для конфигурирования одного чипа для использования со встроенным логическим анализатором SignalTap II, выполните следующие шаги.

1. На панели **конфигурирование цепи JTAG** в редакторе SignalTap II, выберите подключение, которое вы используете для коммуникации из списка **Устройств**. Если вам нужно добавить ваш загрузочный кабель в список, кликните **Установить** для конфигурирования соединения.
2. На панели **конфигурирование цепи JTAG** кликните **Просмотреть** и выберите **.sof** файл, который содержит совместимость со встроенным логическим анализатором SignalTap II.
3. Кликните **Сканировать цепь**. Операция сканирования цепи пронумерует все JTAG чипы внутри JTAG цепи.
4. В списке **Чипы** выберите чип, в который вы хотите загрузить проект. Список чипов содержит порядковый список всех чипов в цепи JTAG. Все чипы нумеруются последовательно, в соответствии с их позицией в цепи JTAG, с префиксом «@». Например, @1 : EP3C25 (0x020F30DD) показывает чип Cyclone III – первый чип в цепи JTAG, с кодом 0x020F30DD.

5. Кликните на иконку **Программировать чип**.

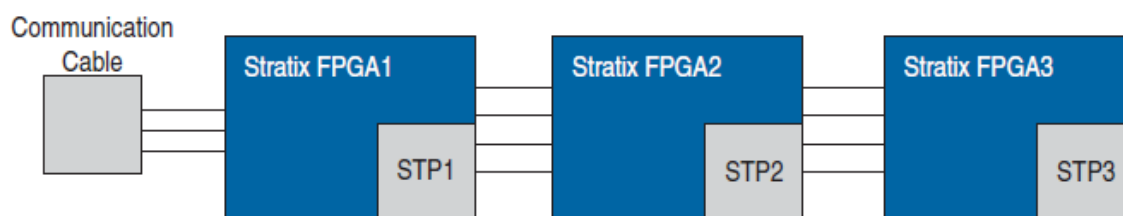
Программирование нескольких чипов для отладки нескольких проектов

Вы можете последовательно отлаживать несколько проектов, используя один элемент в программе Quartus II, выполнив следующие шаги:

1. Создайте, сконфигурируйте и скомпилируйте каждый проект, содержащий **.stp** файл.
2. Откройте каждый **.stp** файл. Не нужно открывать проект Quartus II для открытия **.stp** файла.
3. Используйте контроль панели **конфигурирование цепи JTAG** для выбора нужного чипа в каждом **.stp** файле.
4. Программируйте каждый FPGA.
5. Запустите каждый анализатор независимо.

На рисунке 14-44 показана цепь JTAG и ассоциированные с ней **.stp** файлы.

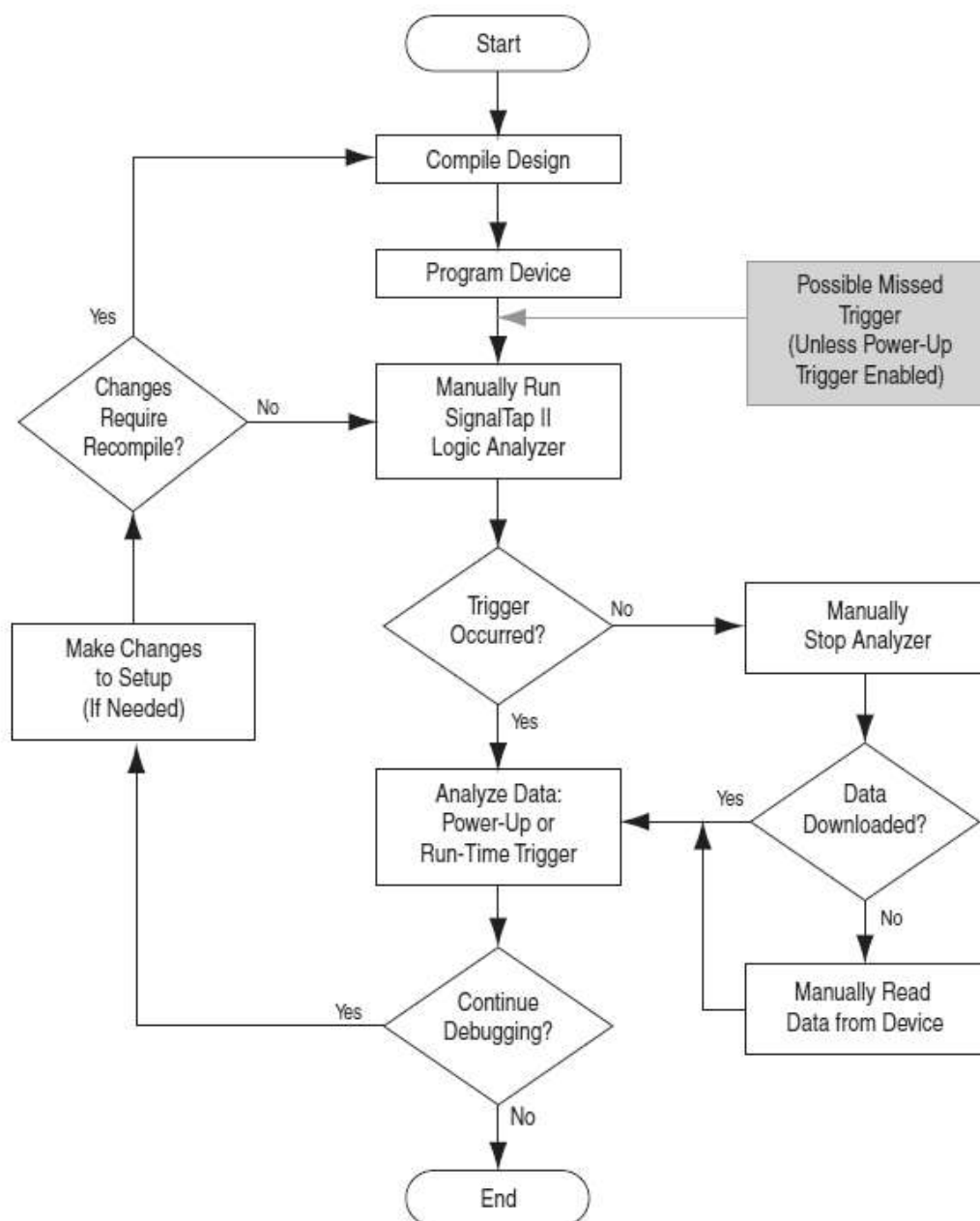
Figure 14-44. JTAG Chain



Запуск встроенного логического анализатора SignalTap II

После того, как чип сконфигурирован вашим проектом, содержащим встроенный логический анализатор SignalTap II, выполняйте операции отладки также как и при использовании внешнего логического анализатора. Вы можете «активизировать» логический анализатор, запустив анализ. Когда происходит событие триггера, захваченные данные сохраняются в буфере памяти в чипе, а затем переносятся в **.stp** файл через JTAG соединение. Вы можете также выполнить эквивалент «форсированного триггера», который позволит вам увидеть текущие захваченные данные без наступления события триггера. На рисунке 14-45 показана блок-схема вашей работы во встроенном логическом анализаторе SignalTap II. На блок-схеме показано, когда происходят события триггера включения и триггера в запущенном времени, и когда захваченные данные от этих событий становятся доступными для анализа.

Figure 14–45. Power-Up and Runtime Trigger Events Flowchart



Панель инструментов в менеджере элементов имеет четыре опции для запущенного анализатора:

- **Запустить анализ** - встроенный логический анализатор SignalTap II запускается до состояния события триггера. Когда происходит событие триггера, мониторинг и захват данных останавливается, если буфер захвата заполнен.

- **Автозапуск анализа** - встроенный логический анализатор SignalTap II последовательно захватывает данные, пока не нажата кнопка «**Стоп анализ**», игнорируя все состояния события триггера.

- **Стоп анализ** – анализ SignalTap II останавливается. Захваченные данные не выводятся автоматически, если событие триггера не происходит.

- **Читать данные** – отображаются захваченные данные. Эта кнопка прекрасно подходит для просмотра захваченных данных, даже если событие триггера не происходит.

Запуск с триггером включения

Если вы разрешили и установили триггеры включения в элемент встроенного логического анализатора SignalTap II, захваченные данные могут быть сразу доступны для наблюдения, если состояние события триггера произошло после конфигурирования чипа. Для загрузки захваченных данных или для проверки, что триггеры включения ещё запущены, в менеджере элемента кликните **запустить анализ**. Если триггер включения сработал, логический анализатор немедленно останавливается, а захваченные данные загружаются в чип. Данные могут сразу же отображаться на вкладке **Данные** редактора SignalTap II. Если триггер включения не сработал, захваченные данные не передаются, а логический анализатор продолжает работать. Вы можете подождать наступления события триггера включения, или остановить логический анализатор, кликнув **стоп анализ**.

Запуск с триггером в запущенном времени

Вы можете активировать и запустить встроенный логический анализатор SignalTap II вручную после конфигурирования чипа для захвата отсчётов данных для триггера в запущенном времени. Вы можете сделать это сразу, если не разрешён триггер включения. Если триггер включения разрешён, вы можете сделать это после того, как данные триггера включения загрузятся из чипа, или если логический анализатор один раз остановится, поскольку событие триггера включения не наступило. Кликните **запустить анализ** в редакторе SignalTap II для начала отображения события триггера. Вы можете запустить несколько элементов SignalTap II одновременно, выбрав все нужные элементы, прежде чем кликнуть **запустить анализ** на панели инструментов.

Пока логический анализатор не будет остановлен вручную, будет начинаться захват данных, когда событие триггера вычисляется как ИСТИНА. Когда это происходит, захваченные данные загружаются из буфера. Вы можете наблюдать данные на вкладке **Данные** редактора SignalTap II.

Выполнение форсированного триггера

Иногда, когда вы используете внешний логический анализатор или осциллограф, вы хотите увидеть текущее состояние сигналов, без установок или ожидания срабатывания триггера. Это относится к операции «форсированный триггер», поскольку вы форсируете тестовым оборудованием захват данных, не взирая на набор состояний триггера. Со встроенным логическим анализатором вы можете выбрать запуск анализатора и прямой захват данных, или запуск анализатора и захват данных по вашему желанию.

Для запуска анализатора и прямого захвата данных, запретите состояния триггера, выключив столбец **Состояния триггера** в списке узлов. Эта операция не требует перекомпиляции. В менеджере элементов кликните **запустить анализ**. Встроенный логический анализатор SignalTap II непосредственно срабатывает по триггеру, захватывает данные и загружает данные во вкладку **Данные** редактора SignalTap II. Если данные не загружаются автоматически, кликните **прочитать данные** в менеджере элементов.

Если вы хотите выбрать ручной захват данных, то запрет состояний триггера не нужен. Для запуска логического анализатора, кликните **Автозапуск анализа**; для захвата данных кликните **Стоп анализ**. Если данные не загружаются во вкладку **Данные** из редактора SignalTap II автоматически, кликните **прочитать данные**.

Вы также можете использовать исходники и пробники в системе совместно со встроенным логическим анализатором SignalTap II для форсирования состояний триггера. Средство исходники и пробники в системе позволяет вам подводить и переключать значения выбранной цепи через цепь JTAG. За дополнительной информацией обратитесь к главе «Отладка проекта с использованием исходников и пробников в системе» в томе 3 Настольной книги Quartus II.

Опции реконфигурации в запущенном времени

Определённые настройки в **.stp** файле можно изменять без необходимости в перекомпиляции, когда вы используете режим триггера в запущенном времени. Все средства реконфигурации в запущенном времени сведены в таблице 14-10.

Таблица 14-10. Средства реконфигурации в запущенном времени

Настройка реконфигурации в запущенном времени	Описание
Базовые состояния триггера и базовые состояния квалификатора памяти	Все сигналы, отмеченные галочкой триггера, могут быть изменены в другое значение базового состояния триггера без перекомпиляции.
Расширенные состояния триггера и расширенные состояния квалификатора памяти	Многие операторы содержат конфигурируемую настройку в запущенном времени. Например, все операторы сравнения конфигурируемые в запущенном времени. Конфигурируемые настройки показаны на белом фоне в представлении блока. Эти опции реконфигурации в запущенном времени разрешаются через диалоговое окно Свойства объекта .
Переключение между квалификатором памяти и непрерывным захватом	Из режима квалификатора памяти, вы можете просто переключаться в режим непрерывного захвата, без перекомпиляции проекта. Для разрешения этого средства, установите галочку для запретить квалификатор памяти .
Параметры процесса базовых состояний триггера	Обратитесь к таблице 14-5 за списком реконфигурируемых опций процесса триггера базовых состояний.

Опции реконфигурации в запущенном времени потенциально сберегают время отладочных циклов, позволяя вам охватывать больше возможных сценариев для элементов, без необходимости перекомпиляции проекта. Компромиссом является незначительное влияние на характеристики и использование ресурсов ядром SignalTap II IP. Реконфигурация в запущенном времени для расширенных состояний триггера и для процесса триггера базовых состояний может быть выключена, для повышения характеристик и уменьшения задействованных ресурсов.

Вы можете сконфигурировать **.stp** файл для предотвращения изменений, которые обычно нуждаются в перекомпиляции. Чтобы сделать это, на вкладке **Установки**, выберите **Позволить только изменения состояний триггера** рядом со списком узлов.

На примере 14-4 показано возможное использование этого случая для средств в запущенном времени. Этот пример содержит описание процесса базовых состояний триггера с разрешённым квалификатором памяти и даёт вам возможность изменять размер окна захвата в запущенном времени без перекомпиляции. Этот пример даёт вам функциональный эквивалент сегментного буфера с одним состоянием триггера, когда размер сегмента реконфигурируем в запущенном времени.

Example 14–4. Trigger Flow Description Providing Runtime Reconfigurable “Segments”

```
state ST1:
if ( condition1 && (c1 <= m) )    // each "segment" triggers on condition
                                //1
begin                            // m = number of total "segments"
    start_store;
    increment c1;
    goto ST2;
end

else (c1 > m )                  //This else condition handles the last
                                //segment.
begin
    start_store
    Trigger (n-1)
end

state ST2:
if ( c2 >= n )                  //n = number of samples to capture in each
                                //segment.
begin
    reset c2;
    stop_store;
    goto ST1;
end

else (c2 < n)
begin
    increment c2;
    goto ST2;
end
```

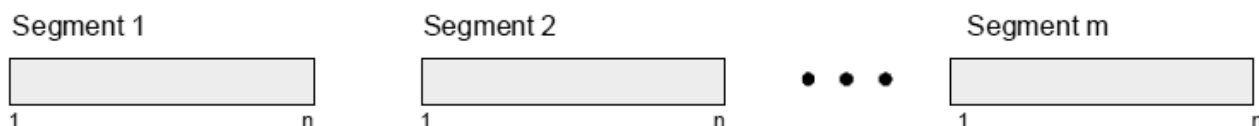
Примечание к примеру 14-1:

(1) $m \times n$ – эквивалент глубины отсчётов, для эффективного использования пространства в этом буфере

На рисунке 14-46 отображается сегментный буфер, описанный в процессе триггеров в примере 14-4.

В запущенном времени, значения m и n могут реконфигурироваться. Изменяя значения m и n в вышеописанном примере, вы можете динамически упорядочить границы сегмента без необходимости перекомпиляции.

Figure 14–46. Segmented Buffer Created with Storage Qualifier and State-Based Trigger (Note 1)



Примечание к рисунку 14-46:

(1) $m \times n$ – эквивалент глубины отсчётов, глубина отсчётов фиксирована

Вы можете добавить состояния в описание процесса триггера и выборочно замаскировать некоторые состояния и разрешить ещё одни в запущенном времени с флагами статуса.

В примере 14-5 показано изменённое описание примера 14-4 с введёнными дополнительными состояниями. Эти специальные состояния используются для определения состояния триггера, который не использует средство квалификатора памяти. Флаги статуса устанавливаются в условные выражения для контроля за выполнением процесса триггера.

Example 14-5. Modified Trigger Flow Description of Example 14-4 with Status Flags to Selectively Enable States

```
state ST1 :

if (condition2  && f1)                                //additional state added for a non-segmented
                                                        //acquisition   Set f1 to enable state
begin
    start_store;
    trigger
end

else if (! f1)
    goto ST2;


state ST2:
if ( (condition1 && (c1 <= m)  && f2)    // f2  status flag used to mask state.  Set f2
                                        //to enable.
begin
    start_store;
    increment c1;
    goto ST3:
end

else (c1 > m )
    start_store
    Trigger (n-1)
end


state ST3:
if ( c2 >= n)
begin
    reset c2;
    stop_store;
    goto ST1;
end

else (c2 < n)
begin
    increment c2;
    goto ST2;
end
```

Статусные сообщения SignalTap II

В таблице 14-11 описаны текстовые сообщения, которые появляются в индикаторе статуса SignalTap II в менеджере элемента перед, во время и после захвата данных. Используйте эти сообщения для информации о состоянии логического анализатора или о том, какую операцию он выполняет.

Таблица 14-11. Текстовые сообщения в статусном индикаторе SignalTap II

Сообщение	Описание сообщения
Not running	Логический анализатор SignalTap II не запущен. Это означает, что нет подключенных чипов или чипы не сконфигурированы.
(Power-Up Trigger) Waiting for clock (1)	Логический анализатор SignalTap II выполняет захват триггера включения или в запущенном времени и ожидает тактовый сигнал для перехода.
Acquiring (Power-Up) pre-trigger data (1)	Состояние триггера было не определено. Был собран полный буфер данных, если использовался режим несегментного буфера, а тип квалификатора памяти – непрерывный.
Trigger In conditions met	Произошло состояние входного триггера. Логический анализатор SignalTap II ожидает появления первого состояния триггера. Это может происходить при определённом состоянии входного триггера.
Waiting for (Power-up) trigger (1)	Логический анализатор SignalTap II ещё ожидает появления состояния триггера.
Trigger level <x> met	Произошло X состояние триггера. Логический анализатор SignalTap II ожидает появления X+1 состояния триггера.
Acquiring (power-up) post-trigger data (1)	Произошли все состояния триггера. Логический анализатор SignalTap II получает данные пост-триггера. Величина данных пост-триггера определяется пользователем между 12%, 50% и 88%, когда выбран режим несегментный буфер захвата.
Offload acquired (Power-Up) data (1)	Данные начали передаваться в программу Quartus II по цепи JTAG.
Ready to acquire	Логический анализатор SignalTap II ожидает запуска анализа пользователем.

Примечание к таблице 14-11:

(1) Это сообщение появляется для триггеров включения или в запущенном времени. Когда обращаются к триггеру включения, появляется текст в круглых скобках.

В режиме сегментного захвата, пре-триггер и пост-триггер не применяется.