

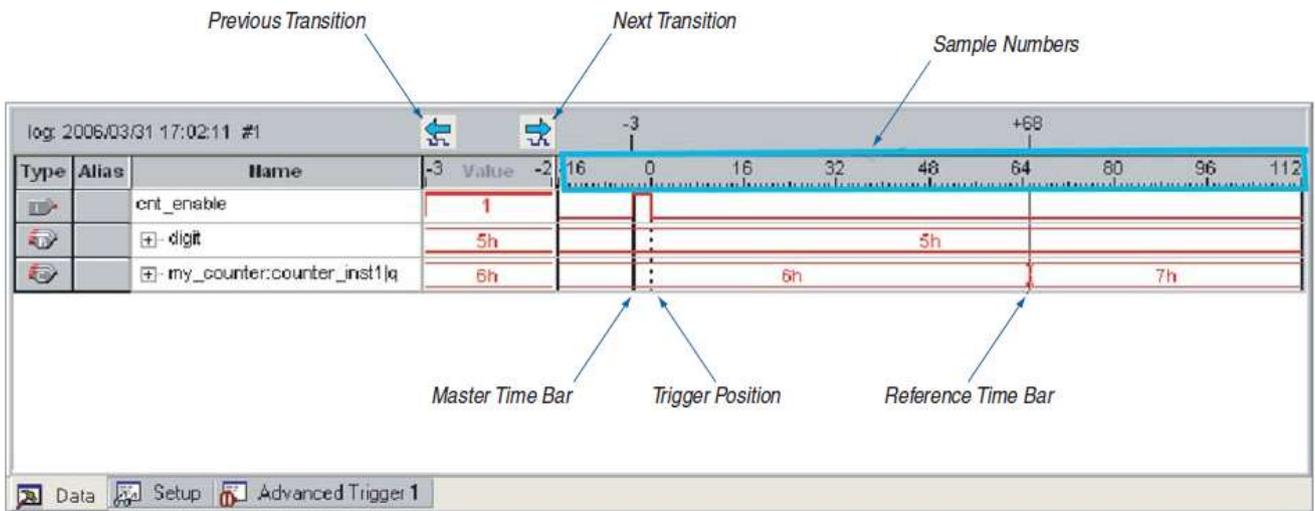
## ***Наблюдение, анализ и использование захваченных данных***

Когда сработал триггер или вы захватили данные вручную, вы можете использовать интерфейс SignalTap II для изучения данных и использования своих находок в отладке вашего проекта.

### **Наблюдение захваченных данных**

Вы можете наблюдать захваченные данные SignalTap II на вкладке **Данные .stp** файла (рисунок 14-47). Каждая строка на вкладке **Данные** показывает захваченные данные одного сигнала или шины сигналов. Шины могут быть развёрнуты, чтобы посмотреть каждый конкретный сигнал в шине. Кликните на временные диаграммы данных для увеличения отсчётов захваченных данных; правый клик для уменьшения.

Figure 14–47. Captured SignalTap II Data



Когда просматриваются захваченные данные, можно узнать временной интервал между двумя событиями. Временной курсор позволяет вам увидеть количество тактовых циклов между двумя отсчётами захваченных данных вашей системы. Существует два курсора времени:

- **Главный курсор времени** – метки главного курсора времени показывают его абсолютное время, выделяются полужирным. Главный курсор времени – это толстая чёрная линия на вкладке **Данные**. Захваченные данные имеют только один главный курсор времени.

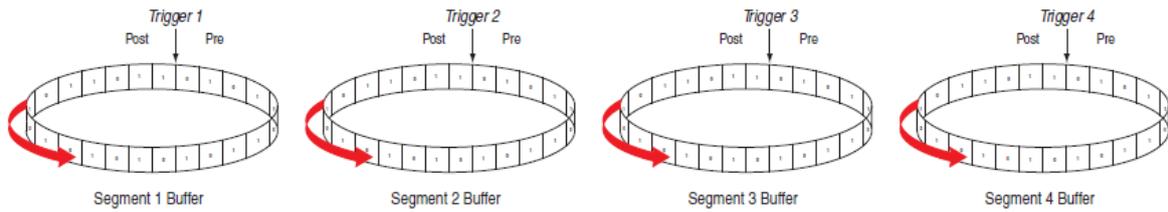
- **Связанный курсор времени** - метки связанного курсора времени показывают время по отношению к главному курсору времени. Вы можете создать неограниченное количество связанных курсоров времени. Чтобы находить переходы сигналов по отношению к главному курсору времени, используйте кнопки **Следующий переход** или **Предыдущий переход**. Это выравнивает положение главного курсора времени по отношению к следующему или предыдущему переходу выбранного сигнала или группы выбранных сигналов. Это средство очень полезно, когда глубина отсчётов очень большая, а размер сигналов переключений очень мал.

Чтобы быстро найти значение шины внутри временной диаграммы, используйте опцию **Найти значение шины**. После того, как вы выбрали шину, правым кликом выберите **Найти значение шины**. Откроется диалоговое окно для ввода параметров поиска.

### Захват данных с использованием сегментных буферов

Сегментные буферы захвата позволяют вам выполнить множество захватов для разных состояний триггера в каждом сегменте захвата. Это средство позволяет вам захватывать по периодическим событиям или последовательности событий, происходящих через длительный период времени. Каждый сегмент захвата работает как несегментный буфер, непрерывно захватывающий данные после активации триггера. Когда вы запускаете анализ с разрешённой опцией **сегментный буфер**, встроенный логический анализатор SignalTap II выполняет один за другим захват данных для каждого сегмента захвата внутри вашего буфера данных. Процесс триггера, или тип и порядок, в котором вычисляются состояния триггера для каждого буфера, определяется либо контролем процесса последовательного триггера, либо контролем процесса настраиваемого триггера базовых состояний. На рисунке 14-48 показан сегментированный буфер захвата, представленный четырьмя сегментами как четырьмя отдельными несегментированными буферами.

**Figure 14-48.** Segmented Acquisition Buffer

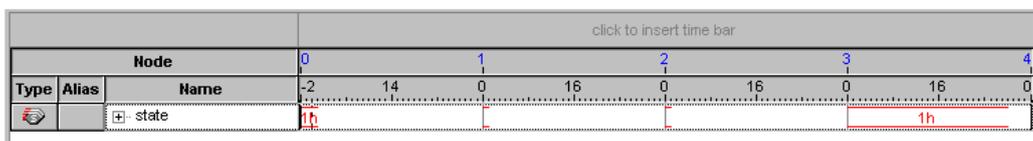


Встроенный логический анализатор SignalTap II заканчивает захват внутри сегмента и начинает новый захват в следующем сегменте. В зависимости от того, какое событие триггера произошло, оно влияет на путь добавления данных в обозреватель временных диаграмм. На рисунке 14-48 показан метод захвата данных. Триггеры, помеченные на рисунке 14-48 как Trigger 1, Trigger 2, Trigger 3 и Trigger 4, обращаются к вычислению команд `segment_trigger` и `trigger` в процессе триггера настраиваемых базовых состояний. Если вы используете последовательный процесс, маркеры триггеров обращаются к состояниям триггера, определённым на вкладке **Установки**.

Если активен первый сегмент буфера, и происходит событие триггера 1, встроенный логический анализатор SignalTap II сразу начинает вычислять триггер 2. Захват данных для второго сегмента буфера начинается после завершения счёта пост-заполнения первого сегмента буфера, или когда вычисляется ИСТИНА для триггера 2, который срабатывает первым. Таким образом, состояния триггера, ассоциированные со следующим буфером при последовательном захвате данных, могут перенять счёт пост-заполнения текущего активного буфера. Это требуется для того, чтобы встроенный логический анализатор SignalTap II мог точно захватывать все происходящие состояния триггера. Отчёты, которые ещё не добавлены, остаются пустым местом в обозревателе временных диаграмм.

На рисунке 14-49 показан пример захвата с использованием последовательного процесса контроля, с установленными состояниями триггера «без разницы». Каждый сегмент до последнего захватывает только один отсчёт данных, поскольку следующее состояние триггера сразу перенимает захват текущего буфера. Позиция триггера для всех сегментов устанавливается пре-триггер (10% данных до состояния триггера и 90% после). Поскольку последний сегмент стартует сразу с состояния триггера, сегмент содержит только данные пост-триггера. Три пустых отсчёта последнего сегмента – это место слева от отсчётов пре-триггера, которое размещено в буфере встроенным логическим анализатором SignalTap II.

**Figure 14-49.** Segmented Capture with Preemption of Acquisition Segments (Note 1)



Примечание к рисунку 14-49:

- (1) Сегментный буфер захвата использует последовательный процесс триггера с установленным состоянием триггера «без разницы». Все сегменты, за исключением последнего сегмента, захватывают только один отсчёт, поскольку следующее состояние триггера закрывает заполнение до конца текущего буфера.

Для последовательного процесса триггера, опция **Позиция триггера** применяется для каждого сегмента в буфере. Для достижения максимальной гибкости в определении позиции триггера, используйте настраиваемый процесс триггера базовых состояний. Упорядочив позиции триггера для вашего сценария отладки, вы можете максимизировать использование выделенного пространства буфера.

### Создание мнемоники для последовательности битов

Средство мнемонические таблицы позволяет вам назначать выразительное имя набору последовательности битов, таких как шина. Для создания мнемонической таблицы, правым кликом на вкладке **Установки** или **Данные .stp** файла, затем кликните **Установка мнемонической таблицы**. Вы создаёте мнемоническую таблицу вводом набора последовательности битов, и определяя метку для представления каждой последовательности. После того, как вы создадите мнемоническую таблицу, правым кликом по группе, кликните **Формат отображения шины** и выберите нужную мнемоническую таблицу.

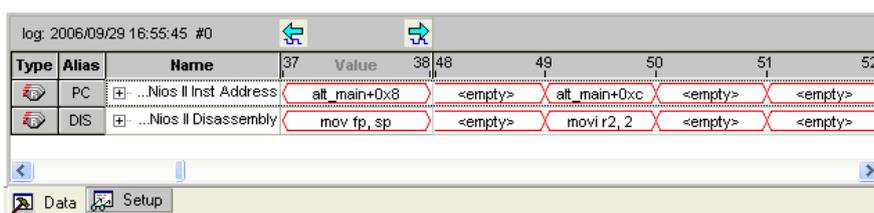
Созданные вами в таблице метки используются двумя способами на вкладках **Установки** или **Данные**. На вкладке **Установки** вы можете создавать базовые триггеры с выразительными именами правым кликом на содержимое в любом столбце **Состояний триггера**, и выбрав метку из таблицы, назначенную вами для группы сигналов. На вкладке **Данные**, если любые захваченные данные создают последовательность битов, имеющуюся в назначенной мнемонической таблице, группа сигналов замещается соответствующей меткой, делая простым наблюдение возникновения последовательности данных.

### Автоматическая мнемоника с плагином

Когда вы используете плагин для добавления сигналов в **.stp** файл, автоматически создаются мнемонические таблицы добавляемых сигналов, которые назначаются сигналам, определённым в плагине. Если вам требуется вручную разрешить эти мнемонические таблицы, правым кликом на имя сигнала или группу сигналов. В подменю **Формат отображения шины** кликните на имя в мнемонической таблице, созданной плагином.

Например, плагин Nios II делает это запросто, для отображения активных сигналов вашего проекта, во время исполнения кода. Если вы установили логическому анализатору защёлкиваться по имени функции в вашем Nios II коде, основываясь на данных из **.elf** файла, вы можете посмотреть имя функции в группе сигналов Адреса элемента по отсчёту триггера, также как и дизассемблерный код в группе сигналов Деассемблера, как показано на рисунке 14-50. Захваченные отсчёты данных вокруг триггера отображаются как смещённые адреса имени функции триггера.

**Figure 14-50.** Data Tab when the Nios II Plug-In is Used



### Локализация узлов в проекте

Когда вы найдёте источник ошибки в вашем проекте, используя встроенный логический анализатор SignalTap II, вы можете использовать средство локализации узлов для локализации сигнала небольшим набором инструментов поиска программы Quartus II, точно также, как в ваших файлах проекта. Это позволяет вам найти источник проблемы быстро, так чтобы вы модифицировали ваш проект, скорректировав дефект. Для локализации сигнала во встроенном логическом анализаторе SignalTap II одним из инструментов программы Quartus II или в ваших файлах проекта, правым кликом на сигнал в **.stp** файле, затем кликните **Локализовать в <имя инструмента>**.

Вы можете локализовать сигнал из списка узлов в любом из следующих мест:

- Редактор назначений
- Планировщик выводов
- Архитектура временного закрытия
- Планировщик чипа
- Редактор свойств ресурсов
- Обзоратель технологической карты
- Обзоратель RTL
- Файл проекта

За дополнительной информацией об использовании этих инструментов, обращайтесь в соответствующие главы Настольной книги Quartus II.

### Сохранение захваченных данных

Журнал данных показывает историю захваченных данных и триггеры, используемые для захвата данных. Анализатор получает данные, сохраняет их в журнале и отображает их в виде временных диаграмм. Когда логический анализатор находится в режиме автозапуска, а состояние триггера наступает неоднократно, захваченные данные для каждого времени события триггера сохраняются в виде отдельных элементов в журнале данных. Это позволяет вам возвращаться назад и просматривать захваченные данные для каждого события триггера. Имя журнала по умолчанию основывается на времени захвата данных. Altera рекомендует вам переименовывать журнал данных более выразительным именем.

Журнал организован в иерархическом виде; похожие журналы захваченных данных сгруппированы вместе в наборе триггеров. Если панель **Журнал данных** закрыта, в меню Вид выберите **Журнал данных** и заново откройте его. Для разрешения ведения журнала, включите **Разрешить журнал данных** во вкладке **Журнал данных** (рисунок 14-25). Для вызова журнала данных заданного набора триггеров, и чтобы сделать их активными, дважды кликните на имя в списке журнала данных.

Средство журнала данных прекрасно подходит для организации различного набора состояний триггера и различного набора конфигурации сигналов. За дополнительной информацией обратитесь к главе «Менеджер нескольких файлов SignalTap II и конфигураций» на странице 14-32.

### Конвертация захваченных данных в другие форматы файлов

Вы можете экспортировать захваченные данные в следующие форматы файлов, некоторые из них могут использоваться другими инструментами симуляции EDA:

- Файл значений, разделённых запятыми (.csv)
- Файл таблицы (.tbl)
- Файл значений дампа изменений (.vcd)
- Файл векторных временных диаграмм (.vwf)
- Файлы графических форматов (.jpg, .bmp)

Для экспорта данных захвата встроенного логического анализатора SignalTap II в меню **Файл** кликните **Экспорт** и выберите **Имя файла**, **Формат экспорта** и **Тактовый период**.

## Создание файла списка SignalTap II

Захваченные данные можно также просматривать в файле списка **.stp**. Файл списка **.stp** – это текстовый файл, содержащий список всех данных, захваченных логическим анализатором по событию триггера. Каждая строка в файле списка соответствует одному отсчёту в буфере захвата. Столбцы соответствуют величине каждого захваченного сигнала или группы сигналов за отсчёт. Если для захваченных данных была создана мнемоническая таблица, числовые значения в списке заменяются созданным содержимым таблицы. Это особенно полезно, когда используется плагин, содержащий инструкции по дизассемблированию кода. Вы можете наблюдать непосредственно порядок, в котором выполняются инструкции кода за один период состояния триггера. Для создания файла списка **.stp**, в меню **Файл** выберите **Создать/обновить** и кликните **Создать файл списка SignalTap II .stp**.