

Использование средства квалификации памяти

Оба буфера – сегментный и несегментный, описанные в предыдущей главе, позволяют делать снимок потока данных во времени для последующего анализа. По умолчанию, задача записи в память захвата с помощью встроенного логического анализатора SignalTap II, - это выборка данных по каждому тактовому циклу. Для несегментного буфера, это означает одно окно данных, в котором представлены состояния снимков потока данных. Аналогично, сегментные буферы используют несколько маленьких окон захвата, разбросанных на продолжительной шкале времени, каждое окно захвата представляет состояния набора данных.

Тщательно выбирайте состояния триггеров и распределение глубины отсчётов для буфера захвата, анализ, использующий несегментные и сегментные буферы, охватывает большинство функциональных ошибок в наборе выбранных сигналов. Поэтому, каждое окно данных может иметь чрезмерную лишнюю избыточную ассоциативность с ними; например, захват потока данных может иметь длительные периоды бесполезных сигналов между посылками данных. Использование настроек по умолчанию для встроенного логического анализатора SignalTap II – это не правильный путь для отбрасывания лишних битов отсчётов.

Средство квалификации памяти позволяет вам отфильтровать конкретные отсчёты, не относящиеся к отлаживаемому проекту. С помощью этого средства, создаётся состояние разрешения записи в буфер для каждого тактового цикла во время захвата данных. После окончательной настройки, полезные данные сохраняются в памяти захвата, средство квалификации памяти позволяет вам более эффективно использовать память захвата и покрывать значительную временную шкалу.

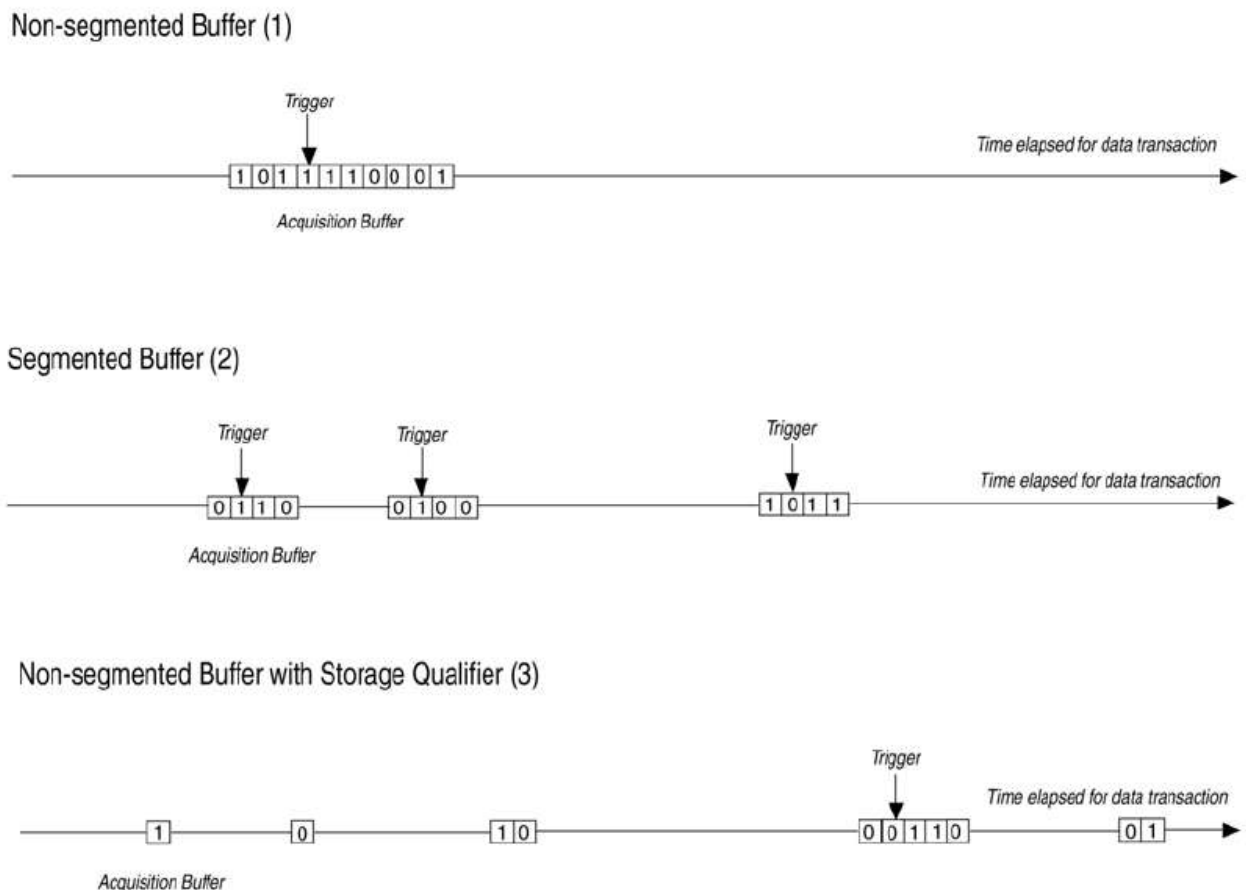
Использование средства квалификации памяти упрощает захват данных с использованием сегментных буферов, из которых вы можете создавать прерывистый буфер захвата. Так как вы можете создавать прерывание между некоторыми двумя отсчётами в буфере, то средство квалификации памяти эквивалентно созданию сегментированного буфера с регулируемыми количеством и размером границ.

На рисунке 14-13 показано три способа записи встроенным логическим анализатором SignalTap II в память захвата данных.

Средство квалификации памяти может быть использовано только для несегментного буфера.

Процесс размещения менеджера плагинов MegaWizard поддерживает только режим Входной порт для средства квалификации памяти.

Figure 14-13. Data Acquisition Using Different Modes of Controlling the Acquisition Buffer



Примечания к рисунку 14-13:

- (1) Несегментные буферы используют фиксированное окно состояний данных.
- (2) Сегментные буферы делят буфер на сегменты с фиксированным размером, каждый сегмент имеет одинаковую глубину выборки.
- (3) Квалификация памяти позволяет вам определить различное окно выборок для каждого сегмента, которое вы создаёте с помощью квалификации состояния. Квалификация памяти потенциально позволяет вам расширить временную шкалу.

Средство квалификации памяти доступно в шести вариантах:

- Непрерывное
- Входной порт
- Переходный
- Условный
- Старт/стоп
- Базовое состояние

Непрерывное (режим по умолчанию) включает средство квалификации памяти. Каждый выбранный тип классификатора памяти активизируется на старте захвата данных. На старте захвата данных, встроенный логический анализатор SignalTap II принимает каждый тактовый цикл и записывает данные в буфер захвата, основываясь на типе квалификации памяти и состоянии. Захват данных останавливается по достижению определённого набора состояний триггера.

Состояния триггера определяются независимо от состоянию квалификатора памяти. Встроенный логический анализатор SignalTap II определяет потоки данных для состояний триггера на каждом тактовом цикле после начала захвата данных.

Состояния триггера определяются в главе «Определение состояний триггера» на странице 14-6. Классификатор памяти работает независимо от состояний триггера. В следующих подразделах описаны все режимы квалификации памяти для буфера захвата данных.

Режим входной порт

Когда используется режим Входной порт, встроенный логический анализатор SignalTap II берёт некоторые сигналы из вашего проекта за входы. Когда проект запущен, если сигнал находится в «1» на фронте такта, то встроенный логический анализатор SignalTap II сохраняет данные в буфер. Если сигнал в «0» на фронте такта, отсчёты данных игнорируются. Внешний вывод создаётся и подключается к этому входному порту по умолчанию, если не определён внутренний узел.

Если вы используете .stp файл для создания элемента встроенного логического анализатора SignalTap II, определите сигнал квалификатора памяти, используя поле входного порта, расположенное на вкладке **Установки**. Этот порт должен быть определён в вашем проекте для компиляции.

Если вы используете процесс менеджера плагина MegaWizard, входной порт квалификации памяти, если указывается, должен добавляться в сгенерированном экземпляре шаблона MegaWizard. Этот порт может быть подключен к сигналу в вашем RTL.

На рисунке 14-14 показано последовательность данных, захваченных сегментным буфером. На рисунке 14-15 показано захват той же последовательности данных, с разрешённым средством квалификации памяти.

Figure 14-14. Data Acquisition of a Recurring Data Pattern in Continuous Capture Mode (to illustrate Input port mode)

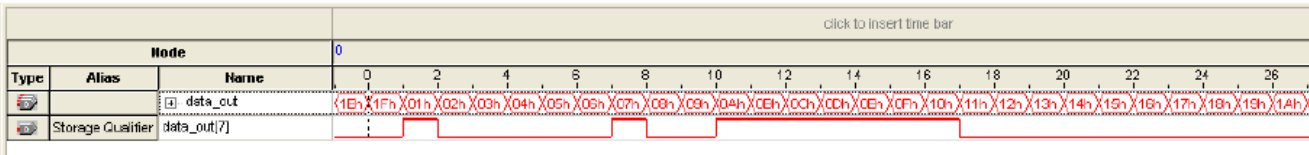
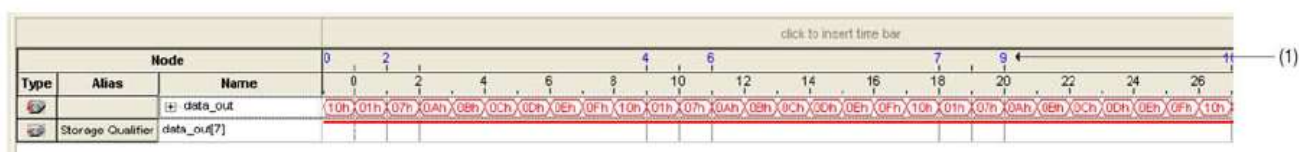


Figure 14–15. Data Acquisition of a Recurring Data Pattern Using an Input Signal as a Storage Qualifier



(1) Маркеры показывают, когда логический анализатор приостанавливает запись в память собранных данных. Эти маркеры разрешаются опцией «Запись данных прерывистая»

Переходный режим

В переходном режиме вы выбираете набор сигналов для контроля, используя вкладку проверки списка узлов в столбце квалификатора памяти. Во время захвата данных, если некоторые сигналы, помеченные для контроля, изменились с момента предыдущего тактового цикла, новые данные записываются в буфер захвата данных. Если нет изменений в помеченных сигналах с момента предыдущего тактового цикла, никакие данные не сохраняются. На рисунке 14-16 показаны установки переходного квалификатора памяти. На рисунках 14-17 и 14-18 показаны захваты последовательности данных в непрерывном режиме и с использованием переходного режима для квалификации данных.

Figure 14–16. Transitional Storage Qualifier Setup

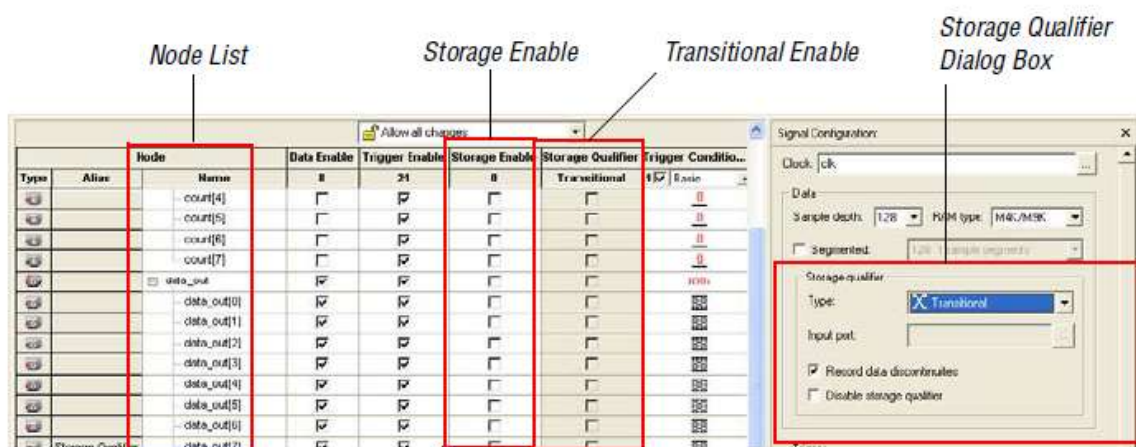
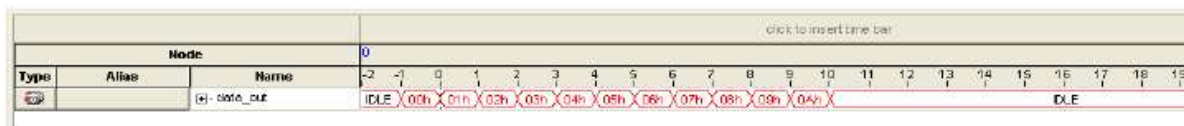


Figure 14–17. Data Acquisition of a Recurring Data Pattern in Continuous Capture Mode (to illustrate Transitional mode)



Условный режим

В условном режиме, встроенный логический анализатор SignalTap II вычисляет комбинационную функцию разрешения сигналов для квалификатора памяти внутри списка узлов для определения, какие данные нужно сохранить. Встроенный логический анализатор SignalTap II записывает в буфер в тот тактовый цикл, в который вы определили состояние как ИСТИНА.

Существуют два вида состояний, которые вы можете определить: базовое и расширенное. Базовое состояние памяти работает с одним из следующих сигналов:

- Без разницы
- Низкий (0)
- Высокий (1)
- Фронт спада
- Любой фронт

Если базовое состояние памяти определено для более, чем одного сигнала, встроенный логический анализатор SignalTap II вычисляет логическое «И» для состояний.

Все другие комбинационные и относительные операторы, которые вы хотите определить как сигнал разрешения для наборов сигналов состояний квалификации памяти, устанавливаются в **.stp** файле.

Вы можете установить состояния квалификации памяти похожим на способ, в котором устанавливаются состояния триггеров. Подробнее о базовых и расширенных состояниях триггеров обратитесь к главам «Создание базовых состояний триггеров» на странице 14-33 и «Создание расширенных состояний триггеров» на странице 14-34. На рисунках 14-20 и 14-21 показан захват данных для анализа в непрерывном режиме, и та же самая последовательность данных с использованием условного режима, соответственно.

Figure 14–19. Conditional Storage Qualifier Setup

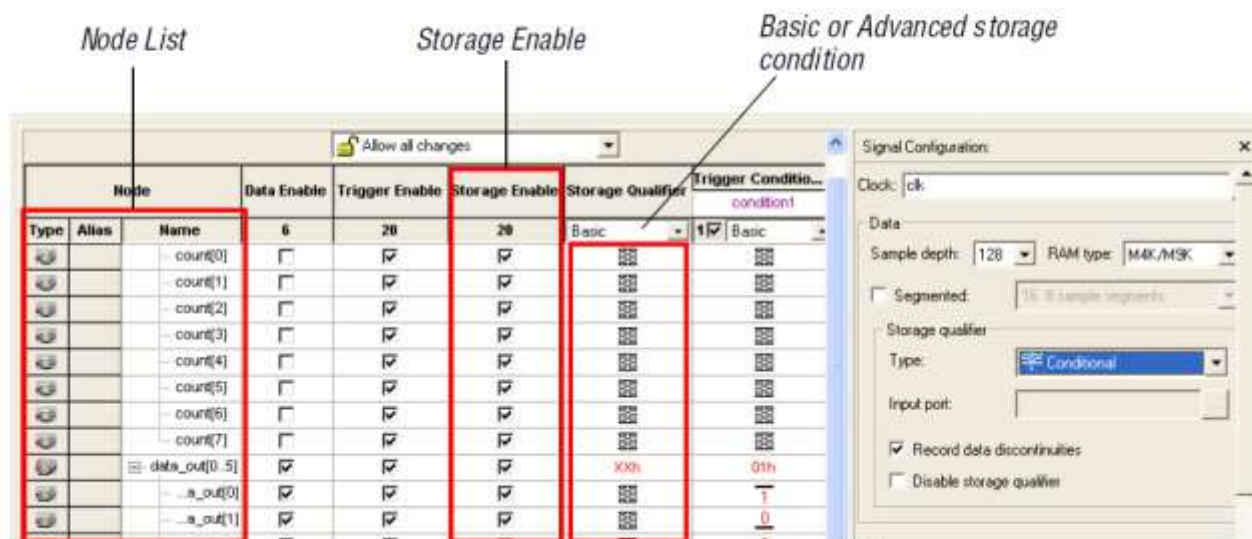
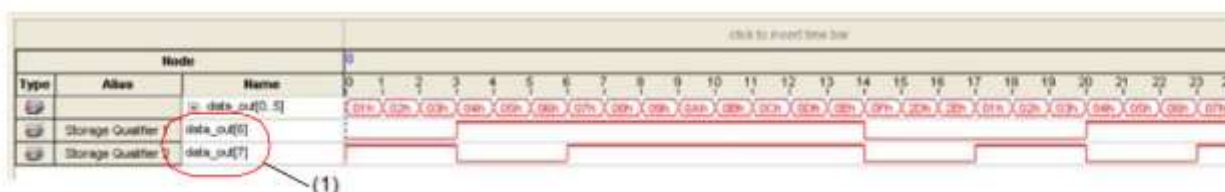
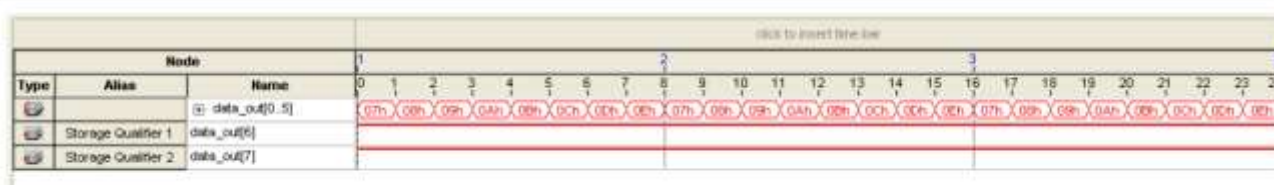


Figure 14-20. Data Acquisition of a Recurring Data Pattern in Continuous Capture Mode (to illustrate Conditional capture)



- (1) Состояние квалификатора памяти приостанавливает захват даны в следующем состоянии: $\text{data_out}[6] \text{ AND } \text{data_out}[7] = \text{True}$. Результат захвата данных квалификатором памяти показан на рисунке 14-21.

Figure 14-21. Data Acquisition of a Recurring Data Pattern in Conditional Capture Mode



Режим старт/стоп

Режим старт/стоп похож на условный режим квалификации памяти. Однако, в этом режиме существуют два состояния: одно для старта и одно для стопа.

Если состояние старта определяется как ИСТИНА, данные начинают сохраняться в буфере по каждому тактовому циклу, пока состояние стопа не определится как ИСТИНА, при этом захват данных приостанавливается. Дополнительные стартовые сигналы, принимаемые после начала захвата данных игнорируются. Если оба сигнала – старт и стоп – определяются как ИСТИНА одновременно, захватывается один тактовый цикл.

Вы можете форсировать триггер для буфера, нажимая кнопку **Стоп**, если буфер вышел из строя, для завершения по состоянию стоп.

На рисунке 14-22 показаны установки режима старт/стоп квалификации памяти. На рисунках 14-23 и 14-24 показаны захват последовательности данных в непрерывном режиме и с использованием режима старт/стоп для квалификатора памяти.

Figure 14-22. Start/Stop Mode Storage Qualifier Setup

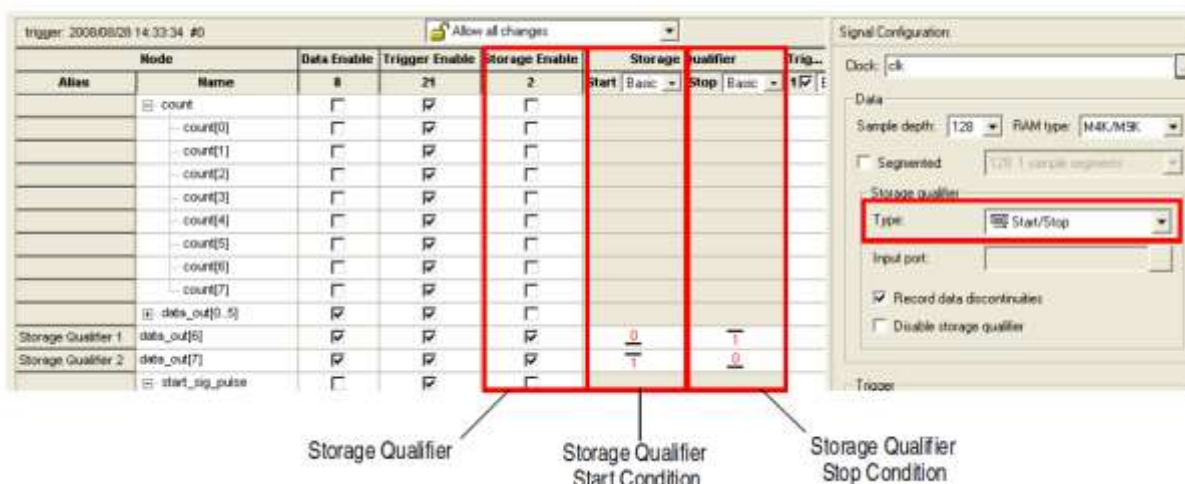


Figure 14-23. Data Acquisition of a Recurring Data Pattern in Continuous Mode (to illustrate Start/Stop mode)

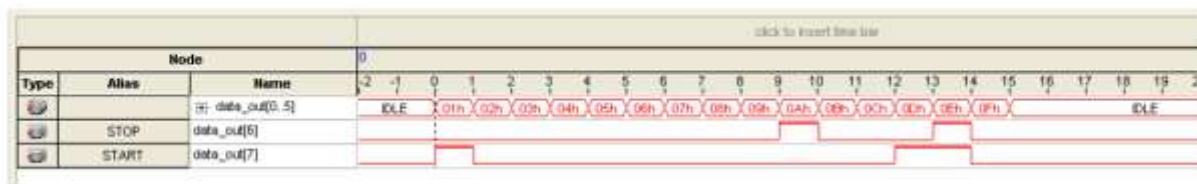
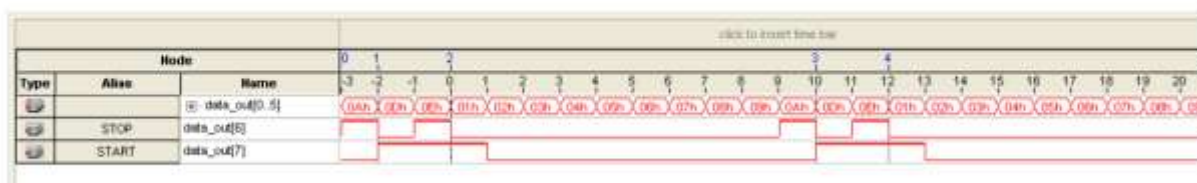


Figure 14-24. Data Acquisition of a Recurring Data Pattern with Start/Stop Storage Qualifier Enabled



Базовое состояние

Режим квалификации памяти базовое состояние используется в процессе переключений базовых состояний. Процесс переключений базовых состояний определяет на основе языка *если-то*, чтобы определить, какие данные будут записаны в буфер. В процессе переключений базовых состояний вы можете командовать булевыми операторами и операторами отношений, чтобы руководить процессом работы выбранного буфера захвата данных. Когда средство квалификации памяти разрешено для процесса базовых состояний, становятся доступными две команды: `start_store` и `stop_store`. Эти команды работают похоже на состояния захвата старт/стоп, описанные в предыдущей главе. После старта захвата данных, данные не записываются в буфер, пока не выполняется команда `start_store`. Команда `stop_store` приостанавливает захват данных. Если обе команды `start_store` и `stop_store` выполняются в один тактовый цикл, один отсчёт сохраняется в буфере захвата.

За дополнительной информацией о процессе базовых состояний и квалификации памяти с использованием процесса переключений базовых состояний, обратитесь к главе «Различные базовые состояния переключений» на странице 14-38.

Показ прерываний данных

Когда разрешается опция во вкладке проверки **Записывать прерывания данных**, встроенный логический анализатор SignalTap II помечает отсчёты, на которых приостанавливается захват средством квалификации памяти. Эти метки показываются в обозревателе временной диаграммы после завершения захвата данных.

Запрещение квалификатора памяти

Во вкладке проверки **Запрещение квалификатора памяти** вы можете просто выключить квалификатор памяти и выполнить непрерывный захват данных. Эта опция может конфигурироваться в реальном времени; это означает, что настройки могут устанавливаться без перекомпиляции проекта. Изменение режима квалификатора памяти в поле **Тип** потребует перекомпиляции проекта.

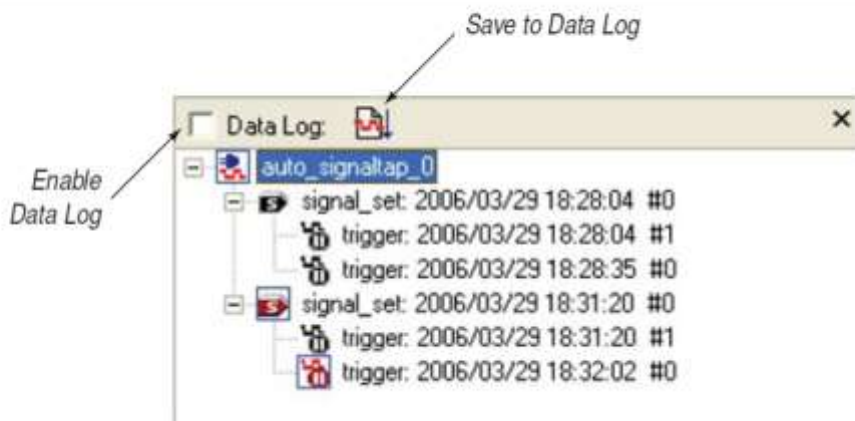
Подробнее о применении опции Реконфигурации в реальном времени во встроенном логическом анализаторе SignalTap II, и о примерах приложений квалификатора памяти, использующих реконфигурируемые опции, обратитесь к главе «Опции реконфигурации в реальном времени» на странице 14-63.

Менеджер нескольких файлов SignalTap II и конфигураций

В некоторых случаях вы можете иметь более одного **.stp** файла в одном проекте. Каждый файл потенциально имеет различные группы мониторинга сигналов. Эти группы сигналов делают возможным отладку различных блоков вашего проекта. В свою очередь, каждая группа сигналов может использовать различные наборы состояний триггеров. Кроме того, каждый **.stp** файл ассоциирован с программным файлом (SRAM файл объекта (**.sof**)). Настройки в выбранном SignalTap II файле должны создавать логический проект SignalTap II в ассоциированном **.sof** файле для логического анализатора, чтобы запускаться непосредственно после программирования чипа. Менеджмент всех **.stp** файлов и ассоциированных с ними настроек, и программных файлов – это сложная задача. Помочь вам управлять всем этим сможет средство **записи данных** и **SOF менеджер**.

Запись данных позволяет вам сохранять различные конфигурации SignalTap II одним **.stp** файлом. На рисунке 14-25 показаны два набора конфигураций сигнала с различными состояниями триггеров в одном **.stp** файле. Для переключения между активными конфигурациями, дважды кликните на них в **Записи данных**. Когда вы переключаетесь между различными конфигурациями, список сигналов и состояний триггеров изменяется во вкладке **Установки .stp** файла. Активная конфигурация, отображаемая в **.stp** файле, показывается в голубом квадрате вокруг набора сигналов в **Записи данных**. Для сохранения конфигурации в **Записи данных**, в меню **Редактировать** кликните **Сохранить в запись данных** или кликните на кнопку **Сохранить** в запись данных вверху **Записи данных**.

Figure 14-25. Data Log



SOF менеджер позволяет вам встроить несколько SOFs в один **.stp** файл. Встраивание SOF в **.stp** файл допускает вам размещение **.stp** файла в других местах: на этом компьютере или в сети, без необходимости иметь ассоциированный **.sof** файл в виде отдельного файла. Для встраивания нового SOF в **.stp** файл, правым кликом в **SOF менеджере**, затем кликните **Прикрепить SOF файл** (рисунок 14-26).

Figure 14-26. SOF Manager



Когда вы переключаетесь между конфигурациями в **Записи данных**, вы можете извлечь SOF, который совместим с индивидуальной конфигурацией, чтобы, используя программатор во встроенном логическом анализаторе SignalTap II, загрузить новый SOF в FPGA. В этом случае, вы контролируете, чтобы конфигурация вашего **.stp** файла всегда создавала проект, программируемый в выбранный чип.