

Определение триггеров

Когда вы запускаете встроенный логический анализатор SignalTap II, он активизирует непрерывный захват данных контролируемых сигналов. «Триггеры» встроенного логического анализатора SignalTap II – означают остановку и отображение данных, когда достигается состояние или набор состояний, которые вы установили. В этой главе описаны различные типы состояний триггеров, которые вы можете установить, используя встроенный логический анализатор SignalTap II.

Создание базовых состояний триггера

Наиболее простое состояние триггера – это базовый триггер. Выберите его из списка сверху столбца **Состояния триггера** в списке узлов Редактора SignalTap II. Вместе с установкой типа триггера *Базовый*, установите структуру триггеров для каждого сигнала, который вы хотите добавить в **.stp** файл. Для установки структуры триггеров, правым кликом в столбце **Состояния триггера** выберите нужный набор. Установите структуру триггеров в одно из следующих состояний:

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

Для шин введите бинарный набор, или правым кликом затем выберите **Вставить значение** и введите структуру в другом числовом формате. Обратите внимание, вы можете ввести «X» для определения значения «без разницы» в любой вашей шестнадцатеричной или бинарной строке. Для сигналов, добавляемых в **.stp** файл, которые имеют ассоциированную мнемоническую таблицу, вы можете правым кликом затем выбрать что-нибудь из таблицы, чтобы установить предопределённое состояние для триггера.

За подробной информацией обратитесь к главе «Наблюдение, анализ и использование данных захвата» на странице 14-66, и в помощи Quartus II.

Для сигналов, добавляемых некоторыми плагинами, вы можете создать базовые состояния, просто используя введённую предопределённую мнемоническую таблицу. Например, для плагина Nios II, если вы имеете определённый **.elf** файл вашего Nios II IDE проекта, вы можете ввести имена функций из вашего Nios II кода. Триггеры логического анализатора переключаются, когда инструкция адреса Nios II обращается по адресу, определённому в коде имени функции.

Захват данных останавливается и данные сохраняются в буфер, когда логическая «И» всех сигналов для нужного состояния триггера устанавливается в значение «ИСТИНА».

Создание расширенных состояний триггера

Вместе с базовыми возможностями триггера во встроенном логическом анализаторе SignalTap II, можете создавать больше комплексных триггеров, использующих дополнительную логику, которые позволят вам захватывать данные при возникновении определённой комбинации состояний. Если вы установили тип триггера **Расширенный** вверху столбца **Состояния триггера** в списке узлов Редактора SignalTap II, добавляется новая вкладка **Расширенный триггер**, в которой вы можете создавать комплексные выражения для триггеров с использованием простой GUI. Для создания комплексного состояния триггера в списке выражений перенесите операторы в окно Редактор расширенной конфигурации триггеров. Для конфигурирования настроек операторов, дважды кликните или правым кликом на операторе, который вы разместили, затем выберите **Свойства**. В таблице 14-4 перечислены операторы, которые вы можете использовать.

Таблица 14-4. Операторы расширенного триггера (1)

Имя оператора	Тип
Меньше	сравнение
Меньше или равно	сравнение
Равно	сравнение
Неравно	сравнение
Больше	сравнение
Больше или равно	сравнение
Логическое «НЕ»	логический
Логическое «И»	логический
Логическое «ИЛИ»	логический
Логическое «ИЛИ-НЕ»	логический
Вычитание «И»	вычитание
Вычитание «ИЛИ»	вычитание
Вычитание «ИЛИ-НЕ»	вычитание
Левый сдвиг	сдвиг
Правый сдвиг	сдвиг
Побитовое дополнение	битовый
Побитовое «И»	битовый
Побитовое «ИЛИ»	битовый
Побитовое «ИЛИ-НЕ»	битовый
Детектирование сигнала по фронту и уровню	детектор

Примечание к таблице 14-4: (1) За дополнительной информацией обратитесь к помощи Quartus II.

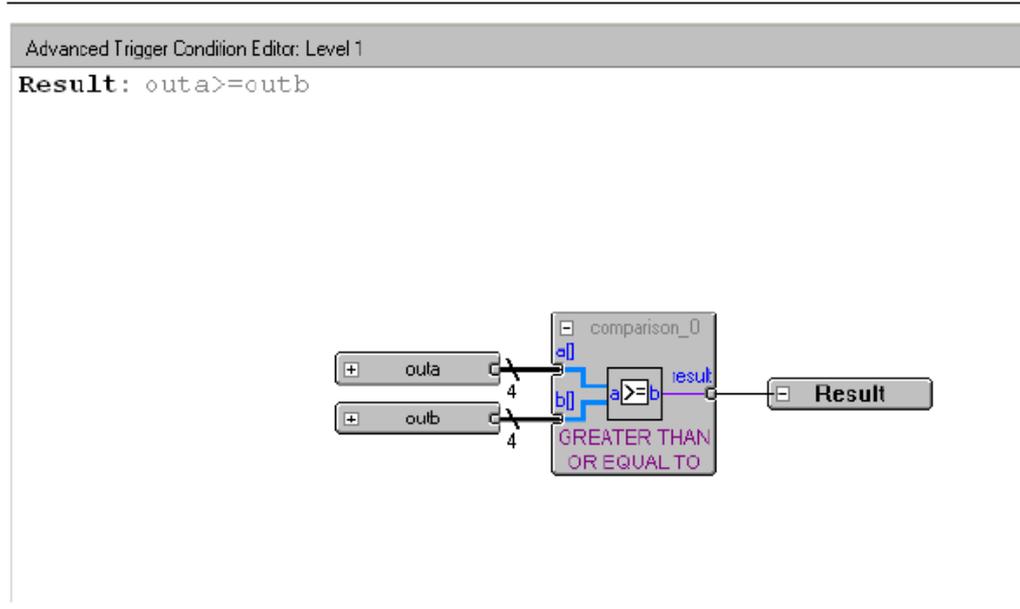
Добавление множества объектов в Редактор расширенных состояний триггера может загромоздить рабочее пространство и усложнить прочтение. Чтобы организовать объекты во время создания расширенных состояний триггера, используйте меню правым кликом и выберите **Систематизация всех объектов**. Вы можете также использовать команду **Уменьшить масштаб**, чтобы разместить больше объектов в окне Редактора расширенных состояний триггера.

Примеры выражений расширенных состояний триггера

В следующих примерах показано, как использовать **Расширенных триггер**:

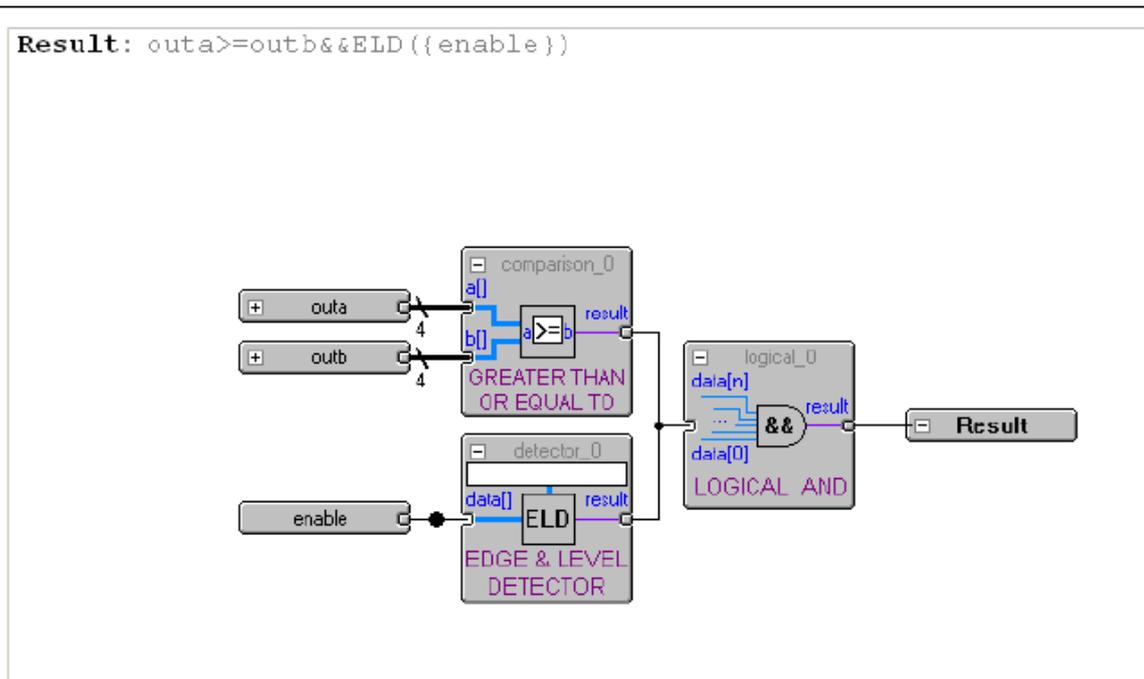
- Триггер, когда шина *outa* больше или равна *outb* (рисунок 14-27).

Figure 14-27. Bus *outa* is Greater Than or Equal to Bus *outb*



- Триггер, когда шина *outa* больше или равна *outb*, а сигнал *enable* на нарастающем фронте (рисунок 14-28).

Figure 14-28. Enable Signal has a Rising Edge



■ Триггер, когда шина *outa* больше или равна *outb*, а сигнал *enable* на нарастающем фронте. Или когда в результате побитовой операции «И» на шинах *outc* и *outd* все биты равны 1 (рисунок 14-29).

Figure 14–29. Bitwise AND Operation

Result: `outa>=outb || ELD({enable}) || (&outc&outd)`

