

# Средства системной отладки САПР Quartus II

**Александр АНТОНОВ**  
antonov@eda-lab.ftk.spbstu.ru  
**Алексей ФИЛИПPOB**  
filippov@eda-lab.ftk.spbstu.ru  
**Роман ЗОЛУТОХO**  
roman@efo.ru

**Данный обзор посвящен средствам отладки САПР Quartus II, позволяющим осуществлять системную отладку проектов на основе СБИС ПЛ производства компании Altera (предполагается, что читатель знаком с базовыми функциями пакета Quartus II).**

Совершенствование архитектуры СБИС программируемой логики и развитие технологии их производства позволяют использовать эти микросхемы для реализации все более сложных проектов. В соответствии с принятым для СБИС ПЛ маршрутом проектирования завершающим этапом процесса разработки является верификация<sup>1</sup> проекта на физической модели, или системная отладка, то есть отладка на плате в окружении реальных устройств, с которыми взаимодействует СБИС ПЛ. Такая отладка позволяет проверить работоспособность устройства при подаче на него реальных сигналов и в условиях помех, устранить нестыковки в интерфейсной части и ошибки разводки печатной платы. Часто исследования на физической модели в реальном времени являются единственным средством верификации системы (особенно для сложных проектов), поскольку имеет ряд причин, ограничивающих возможности программного моделирования (симуляции). Во-первых, достоверность программного моделирования ограничивается соответствием модели входного воздействия реальным условиям. Во-вторых, процесс программной симуляции может занимать недопустимо большое время, так как в модельном времени исследование ведется на несколько порядков медленнее.

Для отладки электронных устройств используются различные приборы: осциллографы, логические анализаторы, генераторы специальных сигналов и др.

Логический анализатор — устройство, предназначенное для записи и анализа циф-

ровых последовательностей. Структура логического анализатора приведена на рис. 1.

Необходимыми компонентами логического анализатора являются:

- память для записи отсчетов;
- средства управления записью в память (цифровая логика);
- средства подключения к наблюдаемым сигналам;
- средства отображения и анализа.

Для записи заданного количества отсчетов последовательности входных цифровых сигналов подается команда начала записи (Trigger). Отсчеты с заданной тактовой частотой записываются в память логического анализатора, а затем выводятся для отображения и анализа.

Ведущие производители измерительного оборудования выпускают логические анализаторы с десятками входов, тактовыми частотами до нескольких сотен мегагерц, имеющие развитые средства отображения и анализа. Существенным ограничением в использовании таких средств является их высокая стоимость (десятки тысяч долларов), сложность подключения к исследуемой системе (особенно к элементам на печатной плате), внесение измерительной погрешности, невозможность контроля внутренних сигналов СБИС ПЛ.

В качестве логических анализаторов могут быть использованы и современные цифровые осциллографы, с той лишь разницей, что они имеют типичное число каналов наблюдения от двух до четырех, а стоимость их находится в пределах нескольких тысяч долларов.

Кроме высокой стоимости у внешних средств наблюдения процессов в цифровых электронных устройствах существует еще один недостаток, связанный со сложностью подключения к интересующим сигналам на плате. Во-первых, подключение приводит

к погрешности измерений. А во-вторых, часто к интересующим сигналам на плате просто нет физического доступа из-за того, что микросхема программируемой логики выполнена в корпусе BGA, а сигналы разведены во внутренних слоях печатной платы.

САПР Quartus II предоставляет комплекс средств для системной отладки проектов, позволяющих как заменить внешние приборы измерения и анализа, так и облегчить подключение этих внешних приборов. К средствам системной отладки САПР Quartus II (рис. 2) относятся:

- редактор отладочных выводов (SignalProbe Pins);
- редактор интерфейса для внешнего логического анализатора (Logic Analyzer Interface Editor);
- редактор содержимого памяти в системном окружении (In-System Memory Content Editor);
- встраиваемый логический анализатор Signal Tap II (Signal Tap II Logic Analyzer).

При использовании средств системной отладки САПР Quartus II рекомендуется выполнять инкрементальную компиляцию<sup>2</sup>. При этом добавление средств отладки не будет вносить изменений в размещение и разводку отлаживаемого проекта и, таким образом, его характеристики останутся неизменными. Кроме того, в случае изменения настроек средств отладки время повторной компиляции будет значительно сокращено.

Пакет Quartus II предоставляет средства для подключения сигналов проекта, доступных в Node Finder после синтеза, к выводам СБИС ПЛ, которые свободны от основных функций проекта и назначены пользователем для контроля функционирования устройства в системе. К таким средствам относятся отладочные выводы (SignalProbe Pins) и интер-

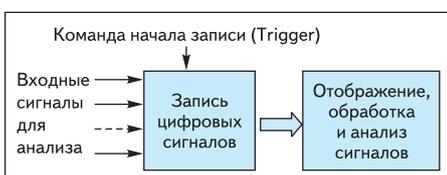


Рис. 1. Структура логического анализатора

<sup>1</sup> Верификация (от лат. verus — истинный, facere — делать) — проверка каких-либо теоретических положений путем их сопоставления с опытными (эмпирическими) данными.

<sup>2</sup> Инкрементальная компиляция позволяет перекомпилировать только те блоки проекта, в которые были внесены изменения со времени предыдущей компиляции.

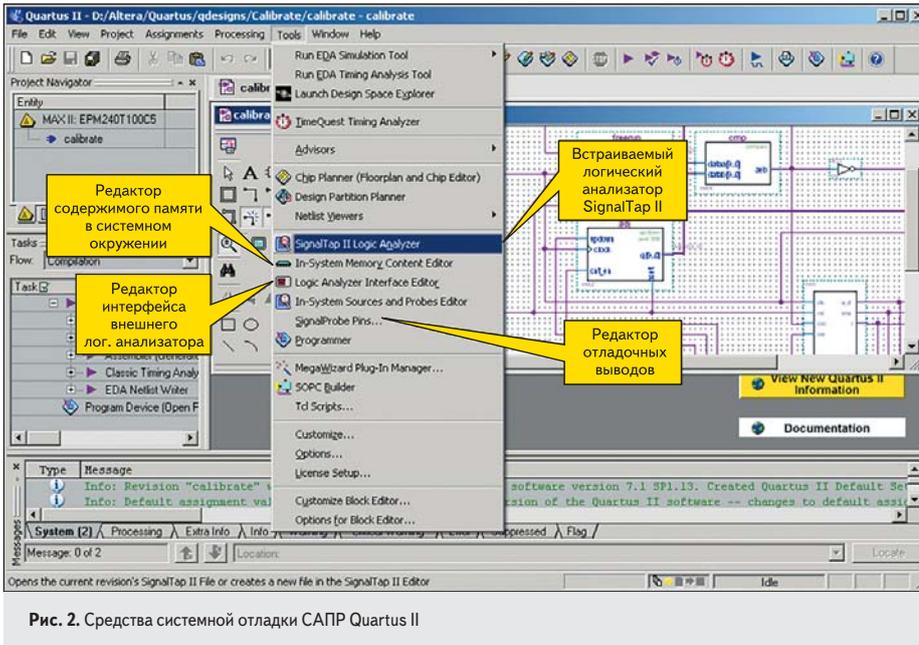


Рис. 2. Средства системной отладки САПР Quartus II

фейс для подключения внешнего логического анализатора (LAI). Выполняя близкие функции по выводу внутренних сигналов проекта на выделенные для отладки контакты СБИС ПЛ, эти средства имеют различные возможности настройки, используют различные редакторы настройки и различный ресурс СБИС ПЛ.

### Отладочные выходы (SignalProbe Pins)

Самым простым средством системной отладки являются отладочные выходы (SignalProbe Pins). Используя SignalProbe, разработчик может сделать доступными для наблюдения внутренние сигналы проекта, назначив их на незанятые в проекте выходы

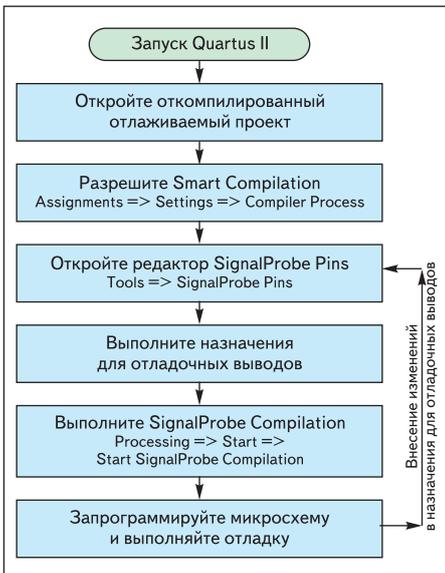


Рис. 3. Последовательность этапов работы с SignalProbe

СБИС ПЛ. Последовательность этапов работы с SignalProbe приведена на рис. 3.

Вид окна редактора отладочных выводов (SignalProbe Pins) показан на рис. 4.

Редактор позволяет выбрать в окне Current and potential SignalProbe pins размещение отладочных выводов (выполняется сортировка по любым колонкам), дать имя выбранному тестовому выводу, указать с помощью Node Finder источник сигнала для него, назначить выход как регистровый или комбинаторный и задать для него стандарт ввода/вывода.

Для добавления требуемых отладочных выводов к проекту нужно разрешить их использование (SignalProbe Enable) перед компиляцией и запустить SignalProbe Compilation. В случае успешной компиляции в колонке Status подключенные отладочные выходы имеют состояние Routed.

При использовании отладочных выводов, создаваемых с помощью SignalProbe, не тратится логический ресурс СБИС ПЛ. Для вывода внутренних сигналов задействуется толь-

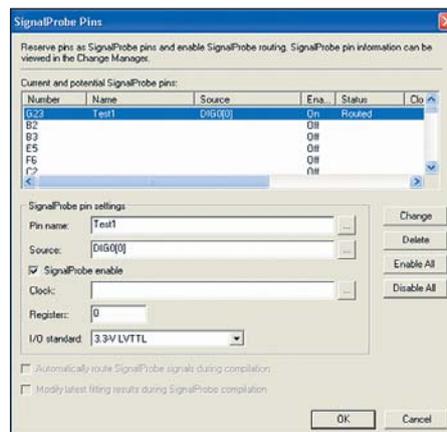


Рис. 4. Окно редактора отладочных выводов

ко неиспользованный в основном проекте ресурс разводки микросхемы. Это гарантирует сохранение характеристик пользовательского проекта.

### Интерфейс для внешнего логического анализатора (Logic Analyzer Interface, LAI)

Упрощенная структура LAI приведена на рис. 5.

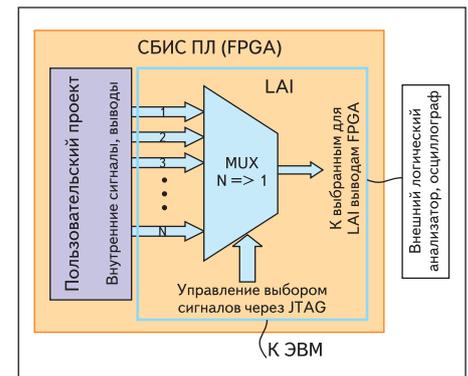


Рис. 5. Упрощенная структура LAI

Каждый тестовый модуль LAI представляет собой параметризуемый по разрядности и числу входов мультиплексор, на вход которого с помощью Node Finder выбираются сигналы анализируемого проекта. Выходы мультиплексора LAI назначаются на незанятые в проекте выходы СБИС ПЛ. Тестируемое устройство через JTAG-интерфейс подключается к компьютеру (с помощью загрузочного/отладочного кабеля). Интерфейс редактора LAI предоставляет пользователю возможность управления мультиплексором для выбора нужных сигналов. Последовательность этапов работы с LAI приведена на рис. 6.

Файл LAI создается посредством редактора пакета Quartus II после завершения ком-

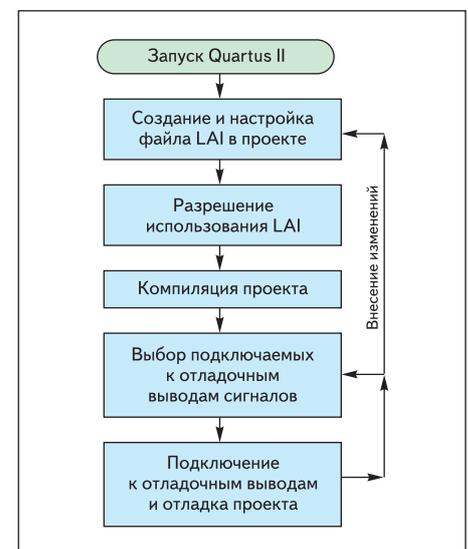


Рис. 6. Последовательность этапов работы с LAI

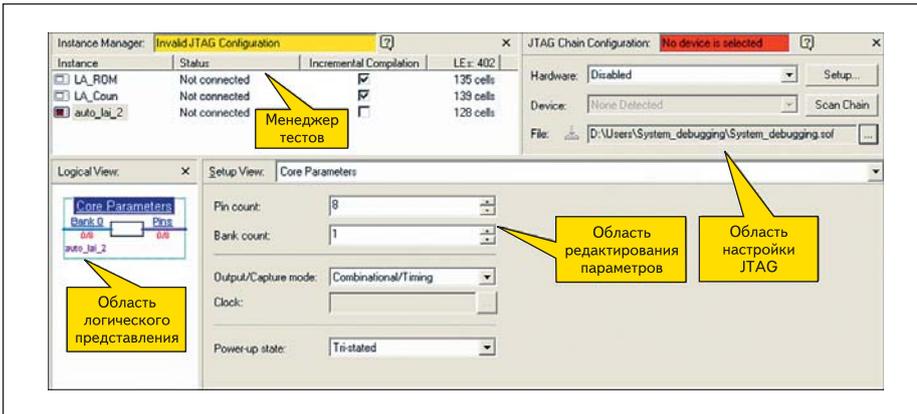


Рис. 7. Окно файла LAI

пиляции проекта (его синтеза, размещения и разводки в кристалле СБИС ПЛ). Прежде чем использовать LAI, рекомендуется провести отладку проекта на программной модели средствами пакета Quartus II.

После создания или изменения интерфейса для внешнего логического анализатора повторяется полная компиляция проекта, при которой средства коммутации, необходимые для LAI, размещаются на свободных функциональных преобразователях СБИС ПЛ. Откомпилированный проект с LAI загружается в СБИС ПЛ, в редакторе LAI для каждого тестового модуля выбираются коммутируемые на выход мультиплексора банки сигналов. Смена банков осуществляется оперативно в редакторе LAI и не требует перекомпиляции и перезагрузки. Поскольку для использования LAI не требуются внутренние блоки ОЗУ СБИС ПЛ, этот интерфейс может размещаться как в микросхемах семейств Arria, Cyclone и Stratix, так и в микросхемах семейства MAX II.

Для открытия нового файла LAI возможны два пути:

- выбрать пункт меню Tools => Logic Analyzer Interface Editor (рис. 2);
- выбрать пункт меню File => New => Other Files => Logic Analyzer Interface File.

В результате откроется окно для настройки нового файла LAI с именем по умолчанию lai1.lai (рис. 7).

Область менеджера тестов (Instance Manager) позволяет в рамках одного файла LAI создать несколько тестовых модулей (но не более 16)

для вывода на контакты СБИС ПЛ внутренних сигналов проекта. Выходные контакты назначаются для каждого тестового модуля индивидуально. Для создания нового теста следует щелкнуть правой кнопкой в поле окна менеджера тестов и выбрать Create Instance. Создаваемым тестам по умолчанию присваивается имя auto\_lai\_x (разработчик может переименовать его по своему усмотрению). Менеджер тестов позволяет выбрать нужный тестовый модуль для настроек и подключения, показывает статус каждого теста и занимаемый аппаратный ресурс.

В области настройки JTAG (JTAG Chain Configuration), не обращаясь к окну программатора, можно:

- задавать средства программирования (в примере на рис. 7 выбор средства программирования еще не сделан);
- определять состояние JTAG-цепи и выбирать нужное для тестирования устройство;
- выбирать программирующий файл и запускать процесс программирования микросхемы СБИС ПЛ.

Область логического представления тестового модуля LAI в виде мультиплексора отображает его настройки. Эти настройки производятся в область редактирования параметров. Здесь для каждого тестового модуля можно задать параметры ядра (Core Parameters), выводов микросхемы (Pins) и групп внутренних сигналов проекта (Banks). Щелчок левой кнопкой в меню выбора настроек области редактирования параметров или по соответствующему имени в обла-

сти логического представления вызывает всплывающее окно настройки выбранного параметра.

Настройка Core Parameters задает:

- количество входов мультиплексора Bank count (присваиваемое по умолчанию имя банка Bank\_x может быть изменено разработчиком по своему усмотрению);
- разрядность выходной шины мультиплексора Pin count;
- комбинаторный или регистровый вывод наблюдаемых сигналов;
- состояние выходов при включении питания.

Настройка выводов (Pins) показывает присвоенные по умолчанию имена выводов тестового модуля LAI и размещение этих выводов на кристалле (рис. 8). Двойной щелчок левой кнопкой в столбце Location открывает окно Pin Planner для задания или корректировки размещения выводов и, при необходимости, стандарта вывода/вывода.

Настройка входов мультиплексора (Banks) выполняется для выбранной группы внутренних сигналов проекта, или для всех групп, и подключает к входным шинам мультиплексора внутренние сигналы, которые будут доступны для наблюдения (рис. 9).

Для каждой группы (банка) заполняется таблица Node, в которой для каждой линии группы с помощью Node Finder (двойной щелчок левой кнопкой в поле Name) выбирается подключаемый внутренний сигнал. В столбце Alias именам выбранных сигналов для удобства восприятия могут быть заданы синонимы.

На рис. 10 приведен пример проекта с использованием LAI. В LAI подключены два созданных тестовых модуля: LA\_ROM и LA\_Coun. Для редактирования параметров выбран тестовый модуль LA\_Coun. В нем для передачи на выход выбран Bank\_0. Назначение банка для передачи на выход мультиплексора возможно только при выполнении двух условий:

- микросхема программируемой логики должна быть сконфигурирована проектом, содержащим LAI;
- отлаживаемое устройство должно быть подключено к управляющему компьютеру (по JTAG-интерфейсу) с помощью загрузочного/отладочного кабеля.

Для подключения другого банка или отключения выводов нужно щелкнуть пра-

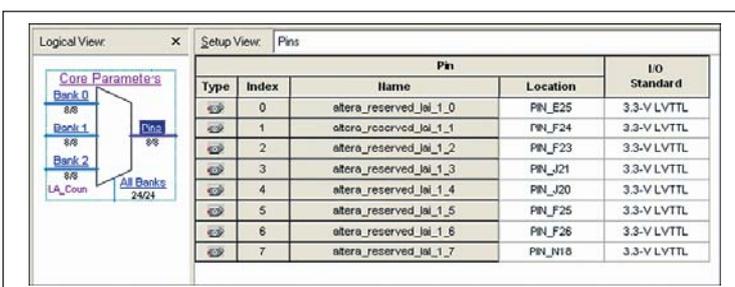


Рис. 8. Настройка выводов LAI

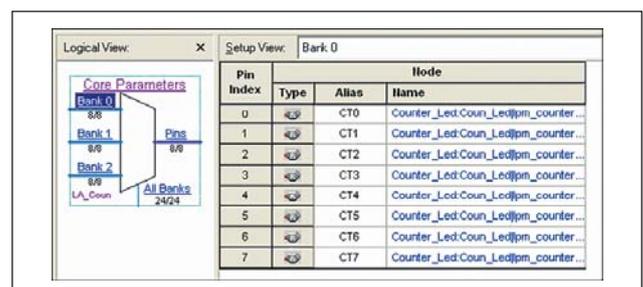


Рис. 9. Настройка входов мультиплексора LAI

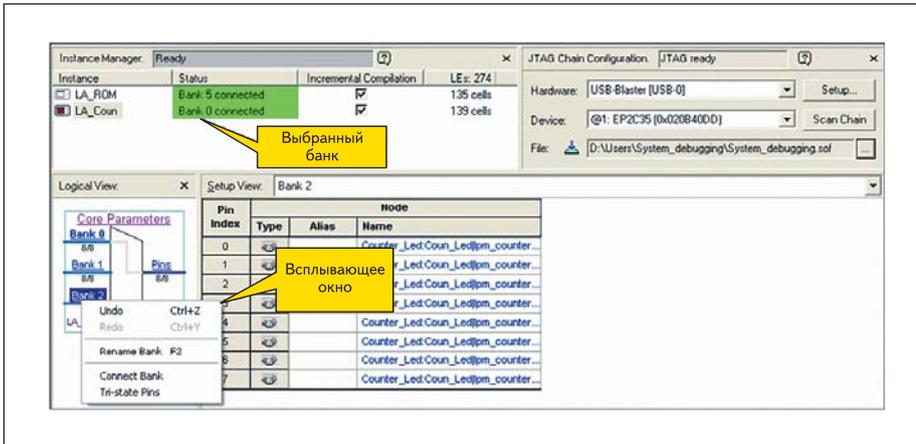


Рис. 10. Пример проекта с использованием LAI

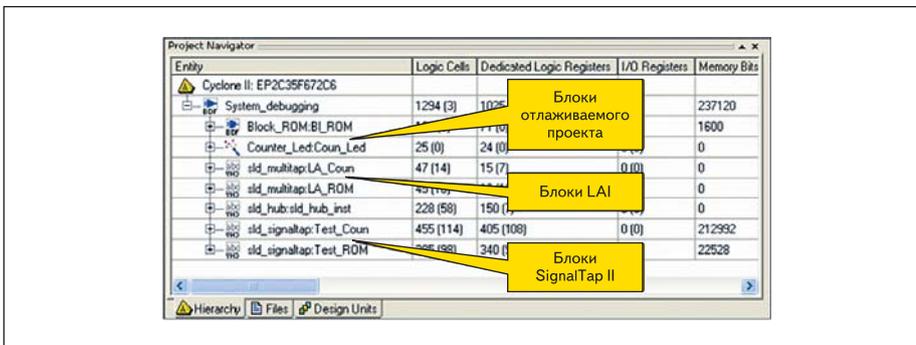


Рис. 11. Окно иерархии проекта с подключенными средствами отладки

вой кнопкой в поле Core Parameters или по наименованию банка. Во всплывающем окне (рис. 10) выполняется требуемая установка.

Прежде чем начать работу с Logic Analyzer Interface, нужно убедиться, что к проекту подключен требуемый файл LAI и что использование интерфейса логического анализатора разрешено. Разрешить использование LAI можно, выбрав пункт меню Assignments => Settings => Logic Analyzer Interface. После завершения процесса отладки интерфейс для подключения логического анализатора может быть исключен из проекта аналогичным образом.

Аппаратные блоки Logic Analyzer Interface автоматически включаются в состав проекта и отображаются в окне иерархии (рис. 11).

### Редактор содержимого памяти в системном окружении (In-System Memory Content Editor)

Анализ данных, записанных в блоки памяти СБИС ПЛ, позволяет упростить отладку систем, в которых содержимое блоков памяти модифицируется в процессе работы (например, проконтролировать содержимое принятого или подготовленного к передаче пакета в системах телекоммуникаций или проанализировать данные, сохраненные в стеке синтезируемого процессорного ядра). Особенно

важна возможность анализа содержимого блоков памяти на физической модели, поскольку симулятор пакета Quartus II не позволяет решить эту задачу на имитационной модели.

Доступ к записанным в память данным в процессе работы осуществляется через создаваемый редактором содержимого памяти в системном окружении (In-System Memory Content Editor) дополнительный порт. Поэтому возможность анализа доступна только для модулей однопортовой памяти — RAM: 1 PORT,

ROM: 1 PORT, и для параметризуемых модулей констант LPM\_CONSTANT. Для анализа в системе содержимого создаваемого блока памяти необходимо указать эту возможность при настройке в редакторе параметризуемых модулей MegaWizard (рис. 12).

Доступ к записанным в настроенную для отладки память данным выполняется с помощью редактора содержимого памяти в системном окружении (рис. 13).

Область настройки JTAG такая же, как и для интерфейса LAI, она позволяет определить положение целевой СБИС ПЛ в JTAG-цепочке, выбрать программирующий файл и выполнить конфигурирование. Для сконфигурированной микросхемы становится активной область менеджера тестирования памяти.

Менеджер тестирования отображает доступные для тестирования блоки памяти и константы с именами, данными им при настройке (рис. 12), их статус и характеристики, позволяет выполнять циклическое или однократное чтение, остановку циклического чтения и запись. Считанные из тестируемых блоков памяти значения отображаются в области данных.

Кроме отображения считанных значений область данных позволяет редактировать данные. Для редактирования стандартным образом выделяется и модифицируется код по выбранному адресу. Щелчок правой кнопкой в поле области данных вызывает всплывающее окно редактора, которое облегчает редактирование, позволяет записать модифицированные слова в память, изменить константы (например, при отладке фильтров), экспортировать считанные данные в файл для дальнейшего анализа и импортировать данные из файла для записи в память. В нижнем правом углу окна отображается положение курсора.

Значения в области данных отображаются:

- черным цветом, если считанные данные совпадают предыдущими отображаемыми данными;

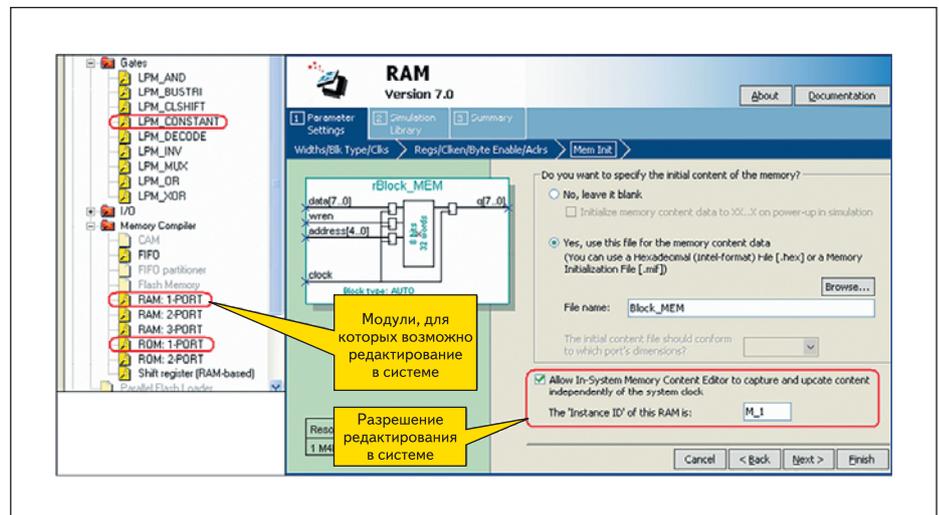


Рис. 12. Настройка параметров блока памяти для работы с In-System Memory Content Editor

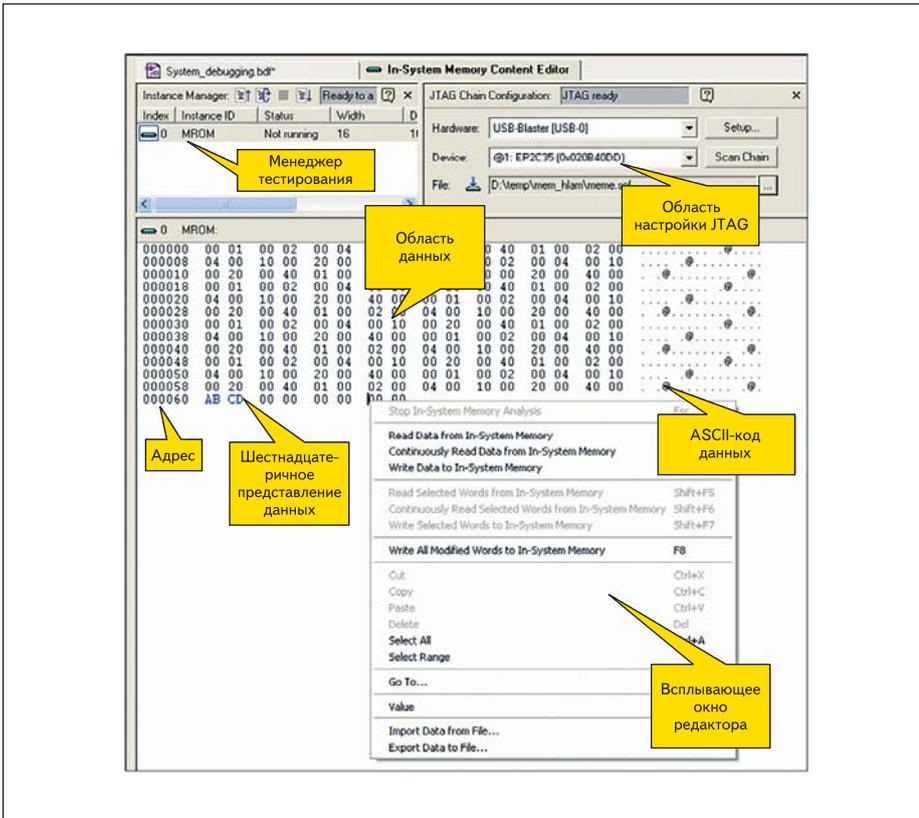


Рис. 13. Окно редактора содержимого памяти в системном окружении

- красным цветом, если считанные данные не совпадают или в СБИС ПЛЛ записаны новые данные из In-System Memory Content Editor;
- синим цветом — модифицированные, но еще не записанные в блоки памяти СБИС ПЛЛ данные.

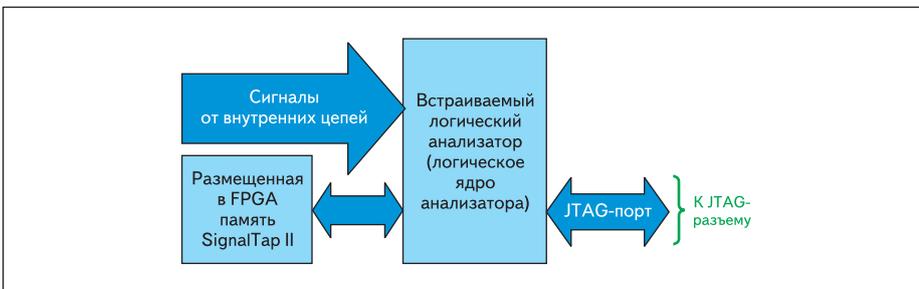


Рис. 14. Функциональная схема SignalTap II

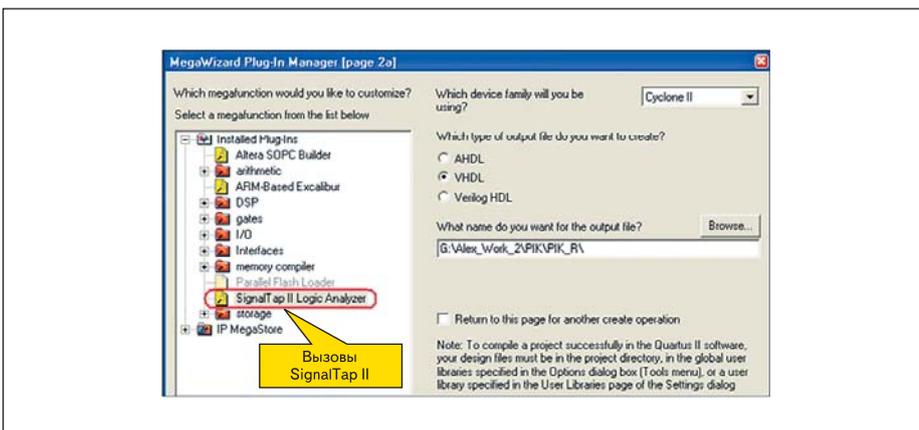


Рис. 15. Вызов мегафункции SignalTap II через MegaWizard

Использование редактора содержимого памяти в системном окружении повышает оперативность отладки и увеличивает ее возможности, так как при этом не требуется переконфигурирования СБИС ПЛЛ для изменения содержимого блоков встроенного ОЗУ.

### Встраиваемый логический анализатор SignalTap II

Наиболее интересным и богатым по возможностям средством системной отладки САПР Quartus II является встраиваемый логический анализатор SignalTap II.

Выше уже были перечислены необходимые компоненты логического анализатора. Все эти компоненты входят в состав системы, включающей отладочную плату со СБИС ПЛЛ:

- память для записи отсчетов в реальном масштабе времени реализуется на блоках встроенного ОЗУ;
- в качестве средств отображения и анализа может использоваться подключенный к СБИС ПЛЛ через загрузочный/отладочный кабель компьютер с соответствующим программным обеспечением;
- управление записью в память может реализовываться на логическом ядре СБИС ПЛЛ;
- подключение к внутренним сигналам и выводам микросхемы СБИС ПЛЛ осуществляется стандартными средствами САПР и вносит минимальные искажения в наблюдаемые сигналы.

Функциональная схема SignalTap II приведена на рис. 14.

Встраиваемый в проект логический анализатор SignalTap II представляет собой параметризируемую мегафункцию, имеющую специализированный пользовательский интерфейс. Доступ к мегафункции SignalTap II возможен как с помощью редактора параметризируемых модулей MegaWizard (рис. 15), так и через специализированный пользовательский интерфейс — открытием str-файла в окне Tools, как показано на рис. 2.

SignalTap II позволяет:

- осуществлять запись логических состояний сигналов проекта, используя выбранный внутренний или внешний сигнал для тактирования;
- выбирать сигналы проекта для наблюдения;
- подключаться к САПР Quartus II через JTAG-интерфейс СБИС ПЛЛ;
- захватывать наблюдаемые сигналы в реальном времени на частотах свыше 300 МГц.

SignalTap II входит как программный модуль в состав Quartus II, в том числе и в его свободную версию Web Edition. Отметим особо, что для использования SignalTap II в Quartus II Web Edition необходимо разрешить опцию TalkBack: Tools => Options => Internet Connectivity => TalkBack Options => Turn on the Quartus II software TalkBack feature.

Основные шаги работы с SignalTap II:

- установка и настройка;

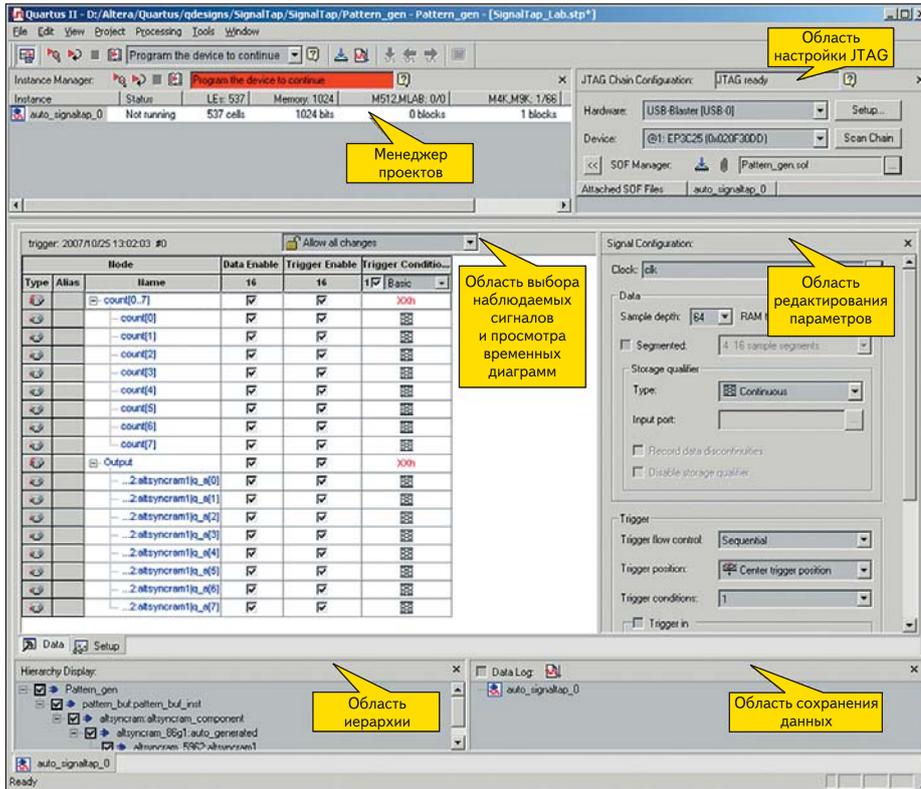


Рис. 16. Окно файла STP

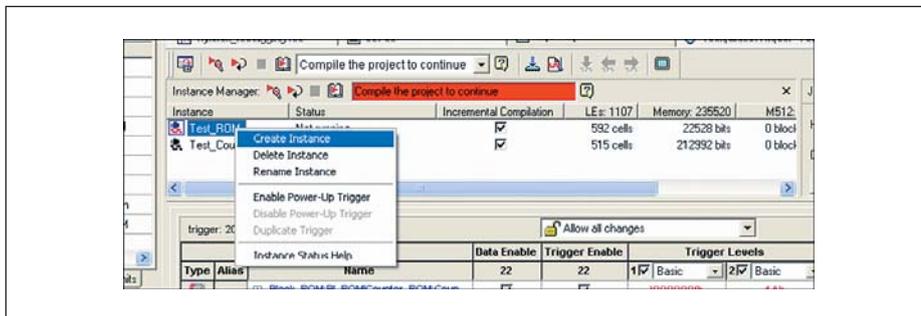


Рис. 17. Создание нового теста STP

- определение момента начала захвата данных (Data Triggering);
- захват данных (Data Capture);
- анализ записанных данных (Data Analysis).

Для проведения отладки проекта могут быть использованы несколько встроенных логических анализаторов (файлов с расширением .stp). Для открытия нового файла STP существуют два способа:

- в меню Tools выбрать пункт SignalTap II Logic Analyzer;
- создать новый файл через меню File => New => Other Files => SignalTap II File.

В обоих вариантах откроется файл с именем по умолчанию STP1.stp (как обычно, разработчик может переименовать его по своему усмотрению). Можно создать несколько разных stp-файлов, но к проекту одновременно может быть подключен только один из них (пункт меню Assignments => Settings => SignalTap II Logic Analyzer => SignalTap II File name).

Пример файла STP, создаваемого через специализированный пользовательский интерфейс, приведен на рис. 16.

- Окно файла STP содержит несколько областей, обеспечивающих настройку тестов, их запуск, отображение и хранение результатов:
- область менеджера тестов (Instance Manager);
  - область редактирования параметров (Signal Configuration);
  - область выбора наблюдаемых сигналов и просмотра временных диаграмм (Waveform Viewer);
  - область настройки JTAG (JTAG Chain Configuration);
  - область иерархии;
  - область сохранения данных.

Область менеджера тестов (Instance Manager) позволяет в рамках одного файла STP создать несколько тестов (но не более 16) для проверки функционирования различных блоков отлаживаемого устройства.

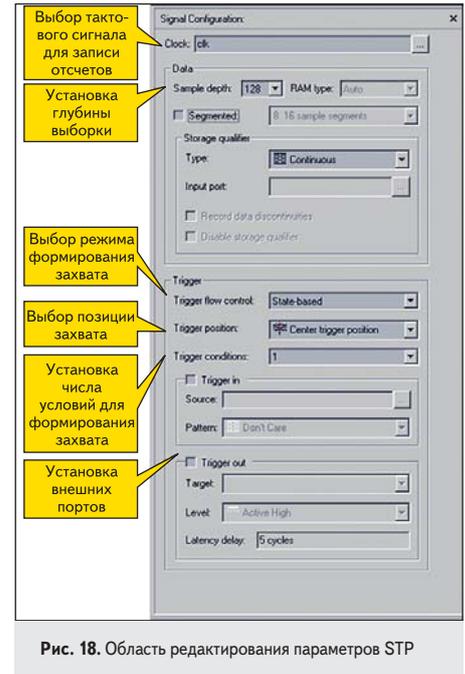


Рис. 18. Область редактирования параметров STP

Для создания нового теста следует щелкнуть правой кнопкой в поле области менеджера тестов (рис. 17) и выбрать Create Instance.

Создаваемым тестам по умолчанию присваивается имя auto\_signaltap\_x, которое в дальнейшем может быть заменено разрабатчиком на содержательное имя. Менеджер тестов позволяет выбрать нужный тест для установок, запуска, отображения, показывает статус каждого теста и занимаемый аппаратный ресурс, указывает на готовность SignalTap II к запуску или необходимость перекompilации проекта при изменениях настроек тестов.

Область редактирования параметров (Signal Configuration) показана на рис. 18.

В этой области можно задать следующие параметры:

- тактовый сигнал для записи отсчетов в память логического анализатора (Sample Clock);
- количество отсчетов в записываемой выборке (Sample Depth);
- режим формирования захвата (Trigger flow control);
- позицию захвата (Trigger);
- число условий для формирования сигнала захвата (Trigger Level).

Выбор сигнала тактирования осуществляется через Node Finder. При выборе тактового сигнала предпочтение должно быть отдано глобальному сигналу. Также может быть выбран внешний сигнал тактирования, с автоматическим созданием внешнего вывода. Запись данных в память логического анализатора осуществляется по каждому положительному перепаду тактового сигнала.

Память для записи отсчетов SignalTap II в зависимости от настроек может быть представлена или в виде кольцевого буфера,

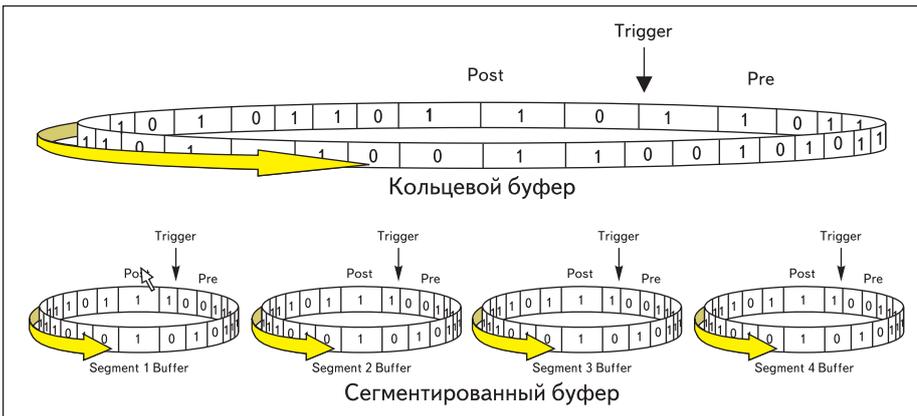


Рис. 19. Представление памяти логического анализатора в виде кольцевого или сегментированного буфера

или в виде сегментированного буфера (рис. 19). В эти буферы непрерывно, с частотой Sample Clock, ведется запись данных. Размер буферов определяется параметром Sample Depth.

SignalTap II имеет два режима формирования сигнала захвата:

- последовательный режим (Sequential) — позволяет использовать и комбинировать стандартные условия захвата;
- режим формирования условий на основе машины состояний (State-based) — позволяет создавать сложные пользовательские условия захвата.

Выбор положения позиции захвата определяет соотношение выводимых отсчетов до позиции Trigger и после нее и осуществляется посредством использования следующих установок:

- Pre — отображаются данные, записанные до момента захвата;
- Center — отображаются данные, записанные как до, так и после момента захвата;
- Post — отображаются данные, записанные после момента захвата.

При использовании сегментированного буфера заданное положение позиции захвата будет одинаковым для всех сегментов.

В качестве сигнала захвата может использоваться и внешний сигнал (порт Trigger-In), для которого задаются условия формирования сигнала захвата.

Для наблюдения внешними инструментальными средствами момента захвата может применяться неиспользуемый в проекте вывод микросхемы (порт Trigger-Out), для которого задается активный уровень сигнала.

Область выбора наблюдаемых сигналов и просмотра временных диаграмм (Waveform Viewer) предоставляет возможности для выбора записываемых в кольцевой буфер сигналов (рис. 20) и формирования условий захвата для заданного количества условий Trigger Level. Также в этой области отображаются записанные в память SignalTap II значения наблюдаемых сигналов (рис. 21) для каждого из создаваемых тестов.

Для выбора сигналов нужно перейти на закладку Setup и двойным щелчком правой клавиши в поле колонки Node/Name вызвать Node Finder. В нем нужно установить маску, позволяющую отображать только те сигналы, наблюдение которых возможно с помощью SignalTap II. Сигналы, наблюдение которых невозможно:

- LE Carry Chain Outputs;
- PLL Clock Output;
- LVDS\_RX/LVDS\_TX;

- CDR\_RX/CDR\_TX;
- сигналы JTAG-интерфейса.

Установленные в окне Setup сигналы могут редактироваться: выбранные шины — разгруппировываться, отдельные сигналы — собираться в шины, для любых сигналов могут задаваться мнемонические таблицы соответствия (например, состояния управляющей шины могут отображаться в виде наименований выполняемых операций, значения семисегментного кода — в виде отображаемых цифр и т. п.).

В колонке Data Enable отмечаются сигналы, которые будут захватываться в конкретном эксперименте — это полезно для уменьшения объема требуемой для записи отсчетов памяти.

В колонке Trigger Enable отмечаются сигналы, которые будут участвовать в логике формирования сигнала захвата (Trigger).

Как уже говорилось, количество условий для формирования сигнала захвата (Trigger Level) задается в области редактирования параметров. Простейший способ формирования сигнала захвата (Trigger) в последовательном режиме — использование базовых условий (Basic), варианты которых предлагаются при щелчке правой кнопкой в колонке Trigger Level соответствующей строки. Каждый Trigger Level выполняется только тогда, когда одновременно выполняются усло-

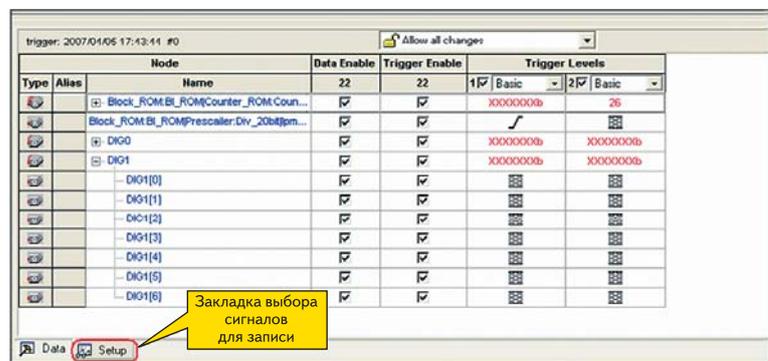


Рис. 20. Выбор записываемых в буфер SignalTap II сигналов

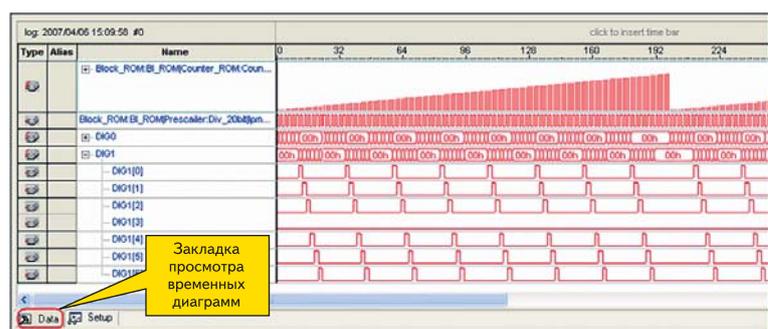


Рис. 21. Просмотр временных диаграмм STP

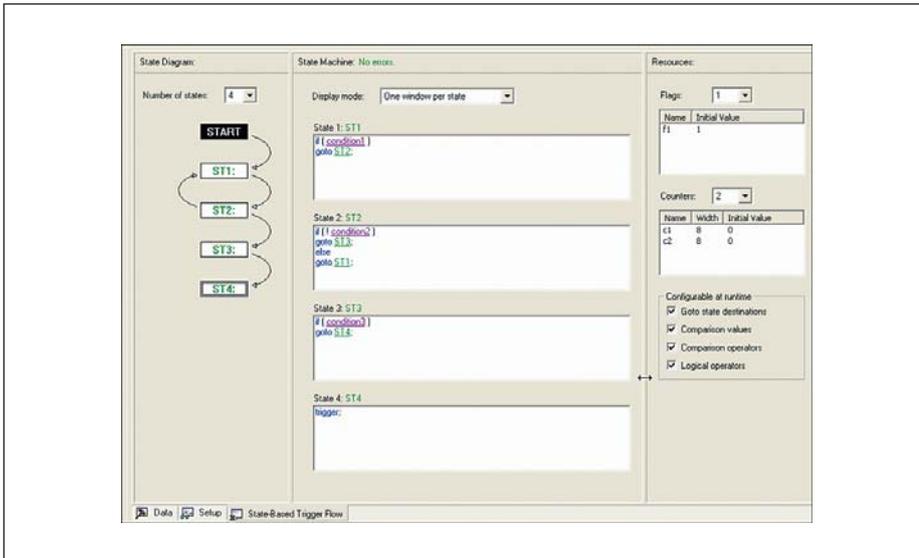


Рис. 22. Формирование условий захвата с использованием машины состояний

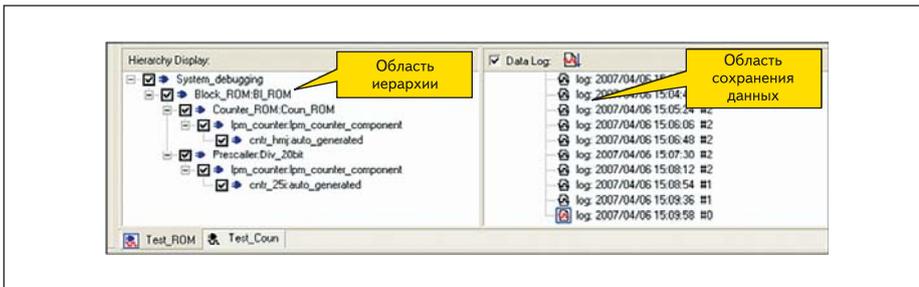


Рис. 23. Области иерархии и сохранения данных STP

вия, заданные для всех сигналов, участвующих в его формировании.

Для формирования более сложных условий захвата в последовательном режиме (например, с использованием логических операторов, операторов отношения и операторов задержки) можно использовать сложные условия (Advanced).

В режиме формирования условий захвата на основе машины состояний разработчик может формировать сложные условия захвата — например,  $n$ -кратное повторение цепочки некоторых событий. В этом режиме процесс формирования захвата описывается в виде машины состояний, для которой разработчик задает условия перехода из состояния в состояние и действия, производимые в каждом состоянии. Для описания машины состояний используется простой BASIC-подобный язык (рис. 22).

Сигнал захвата Trigger формируется в случае выполнения любого из активных условий Trigger Level. После осуществления захвата (в соответствии с установленной позицией захвата — Pre, Center или Post) запись данных в буфер SignalTap II прекращается и содержимое буфера становится доступным для анализа.

Для просмотра временных диаграмм после захвата данных нужно перейти на закладку

Data (рис. 21). Формат отображаемых данных может быть задан во всплывающем меню, вызываемом щелчком правой кнопки на выбранном сигнале.

Область иерархии (Hierarchy Display) (рис. 23) отображает иерархию сигналов, выбранных для просмотра временных диаграмм, и позволяет скрывать любые из них для удобства анализа остальных.

Область сохранения данных (Data Log) (рис. 23) позволяет записывать тестовые последовательности для дальнейшего их просмотра средствами SignalTap II или преобразования в другие форматы (выбор записанного теста в пункте меню Data Log => File => Export => Сохранение в выбранном формате). Преобразованные тестовые дан-

ные могут наблюдаться и анализироваться другими средствами, например, средствами редактора временных диаграмм САПР Quartus II.

Область настройки JTAG (JTAG Chain Configuration), как и для рассмотренных выше средств системной отладки, задает интерфейс компьютера и отлаживаемого устройства (рис. 24).

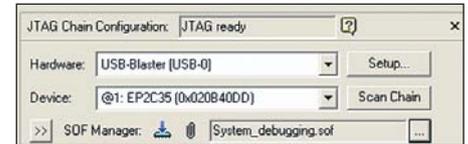


Рис. 24. Окно настройки JTAG

В этой области, не обращаясь к окну программатора Quartus II, можно производить следующие действия:

- задавать средства программирования (в примере на рис. 24 выбран загрузочный/отладочный кабель USB Blaster);
- определять состояние JTAG-цепочки и выбирать нужную для тестирования микросхему;

- выбирать программирующий файл и запускать процесс конфигурации СБИС ПЛ.

После окончания отладки логический анализатор SignalTap II может быть исключен из проекта (пункт меню Assignments => Settings => SignalTap II Logic Analyzer). После удаления SignalTap II из проекта следует сохранить существующую разводку проекта (пункт меню Assignments => Back-Annotate Assignments => Pin, sell, routing and device assignments) и перекомпилировать проект.

Установки всех средств системной отладки, используемых в проекте, отображаются в окне отчета Compilation Report => Analysis & Synthesis => Debug Settings Summary (рис. 25).

Средства системной отладки САПР Quartus II удобны в использовании, и позволяют существенно упростить процесс верификации проектов на основе СБИС ПЛ в реальном аппаратном окружении. Любое из них может быть добавлено к проекту для осуществления физического моделирования и исключено из проекта после его завершения, при переходе от прототипа к серийной продукции. ■

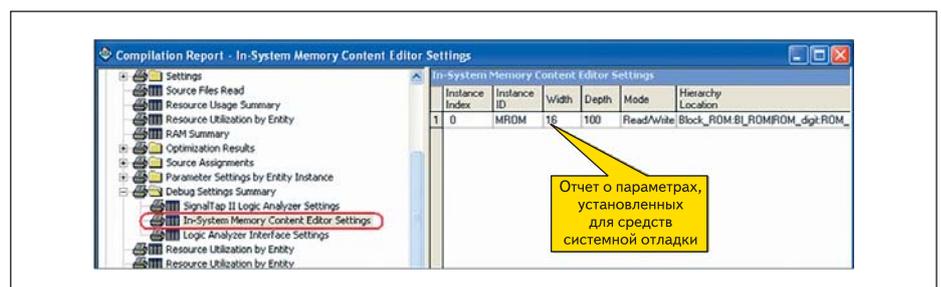


Рис. 25. Отображение установок In-System Memory Content Editor в окне отчета