

Окончание. Начало в № 11 '2008

Формирование описаний компонентов для внутрикристальной отладки цифровых устройств и встраиваемых микропроцессорных систем на основе параметризованных модулей Xilinx CORE Generator Tool

Валерий ЗОТОВ
walerry@km.ru

Четвертая часть завершает описание процедуры подготовки компонентов, генерируемых с помощью параметризованных модулей Xilinx CORE Generator Tool [1], которые предназначены для аппаратной отладки цифровых устройств и встраиваемых микропроцессорных систем, реализуемых на основе ПЛИС с архитектурой FPGA [2]. В данной части рассматривается формирование описаний элементов, предоставляющих возможность применения логических анализаторов компании Agilent Technologies в процессе внутрикристальной отладки.

Формирование описаний элементов, обеспечивающих возможность использования в ходе внутрикристальной отладки разрабатываемых устройств и встраиваемых микропроцессорных систем логических анализаторов компании Agilent Technologies

Чтобы использовать в процессе внутрикристальной отладки разрабатываемых устройств и встраиваемых микропроцессорных систем аппаратные логические анализаторы, выпускаемые компанией Agilent Technologies, необходимо с помощью средств CORE Generator [3] подготовить описание соответствующего элемента на основе параметризованного модуля *Agilent Trace Core 2 (ATC2)*. Ко времени подготовки данной публикации последней версией этого ядра являлась модификация *Agilent Trace Core v1.01a*. Указанная версия рассматриваемого параметризованного модуля предназначена для генерации описаний контроллеров *ATC2*, используемых в процессе внутрикристальной отладки цифровых устройств и микропроцессорных систем, реализуемых на базе ПЛИС следующих семейств: Spartan-3, Spartan-3 XA, Spartan-3E, Spartan-3E XA, Spartan-3A, Spartan-3AN, Spartan-3A DSP, Virtex-II, QPRO Virtex-II Rad Tolerant, QPRO Virtex-II Military, Virtex-II Pro,

Virtex-4 FX, Virtex-4 LX, Virtex-4 SX, QPRO Virtex-4 Rad Tolerant, QPRO Virtex-4 Hi-Rel, Virtex-5 LX, Virtex-5 LXT, Virtex-5 SXT и Virtex-5 FXT.

Основными особенностями параметризованного модуля *Agilent Trace Core 2 (ATC2)* версии v1.01a являются:

- возможность предоставления одновременного доступа к 16 384 контрольным точкам отлаживаемого устройства или встраиваемой микропроцессорной системы;
- организация цепей входных сигналов в виде банков, поддерживающая от 1 до 64 банков;
- возможность выбора количества линий входных сигналов (разрядности) банков в диапазоне от 1 до 64;
- поддержка использования до 128 выходов данных, подключаемых непосредственно к входному разъему Probe логического анализатора компании Agilent Technologies;
- возможность выбора синхронного или асинхронного режима захвата контролируемых сигналов;
- наличие опции временного мультиплексирования, позволяющего удвоить количество линий входных сигналов каждого банка;
- поддержка различных цифровых сигнальных стандартов ввода/вывода на выходах формируемого элемента, подключаемых к логическому анализатору;

- возможность индивидуального выбора стандарта ввода/вывода, нагрузочной способности и скорости переключения фронтов для каждого выхода сигнала, подключаемого непосредственно к логическому анализатору компании Agilent Technologies. Элементы, генерируемые с помощью рассматриваемого параметризованного модуля, выполняют функции виртуального логического анализатора и контроллера интерфейса передачи данных на входы аппаратного анализатора компании Agilent Technologies. При этом в отличие от конфигурируемых логических анализаторов, создаваемых на основе параметризованного модуля *Integrated Logic Analyzer (ILA)*, рассмотренного во второй части данной статьи (см. КиТ № 12 '2008 года), в этих элементах не используются ресурсы блочной памяти ПЛИС Block RAM. Определение условий захвата и фиксации контролируемых сигналов, а также управление их отображением в процессе отладки разрабатываемых устройств и встраиваемых микропроцессорных систем при использовании ядра *Agilent Trace Core 2 (ATC2)* осуществляется с помощью аппаратного логического анализатора компании Agilent Technologies.

В составе «мастера» настройки параметризованного модуля *Agilent Trace Core 2 (ATC2)* версии v1.01a используются от трех до десяти диалоговых панелей, в зависимости от выбранной разрядности банков входных сигна-

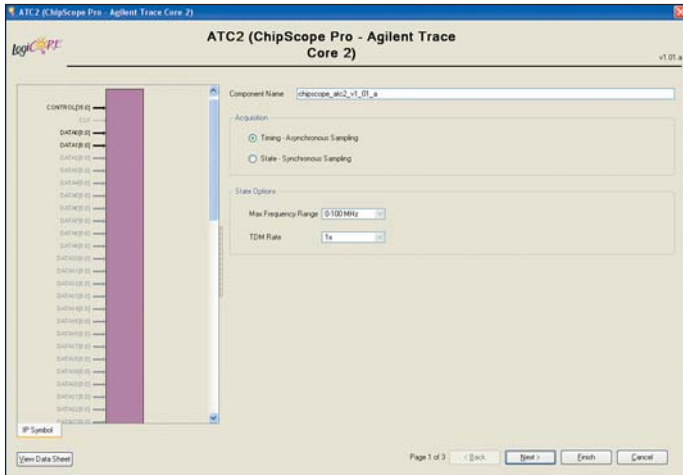


Рис. 17. Вид стартовой диалоговой панели «мастера» настройки параметров ядра Agilent Trace Core 2 (ATC2) версии v1.01a

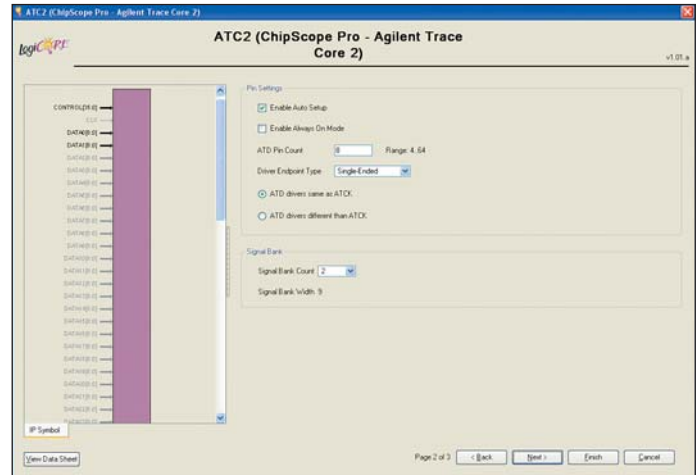


Рис. 18. Вид второй диалоговой панели «мастера» настройки параметров ядра Agilent Trace Core 2 (ATC2) версии v1.01a

лов. Стартовая диалоговая панель этого «мастера», вид которой представлен на рис. 17, предназначена для выбора режима захвата и предполагаемого диапазона частот контролируемых сигналов.

Режим захвата регистрируемых сигналов выбирается с помощью двух кнопок с зависимой фиксацией, расположенных во встроенной панели *Acquisition*. Чтобы установить асинхронный режим захвата контролируемых сигналов, следует зафиксировать в нажатом состоянии кнопку *Timing — Asynchronous Sampling*. В этом случае все сигналы, поступающие на входы данных формируемого элемента, транслируются на соответствующие выходы исключительно через комбинационную логику. При этом выходной контакт АТСК генерируемого компонента используется в качестве дополнительного выхода данных. Для выбора синхронного режима захвата регистрируемых сигналов необходимо переключить в нажатое положение кнопку *State — Synchronous Sampling*. В этом режиме контролируемые данные передаются на выходы через последовательность конвейерных регистров, которые тактируются сигналом синхронизации, поступающим на соответствующий вход CLK формируемого элемента.

При установке синхронного режима захвата следует указать границы предполагаемого диапазона частот контролируемых сигналов, используя поле выбора *Max Frequency Range*, представленное во встроенной панели *State Options* (рис. 17). Параметризованный модуль *Agilent Trace Core 2 (ATC2)* позволяет формировать описания элементов, которые оптимальным образом размещаются и трассируются в кристалле с учетом максимального значения частоты регистрируемых сигналов. Выпадающий список возможных значений параметра *Max Frequency Range* включает в себя четыре варианта: 0–100, 101–200, 201–300 и 301–500 МГц. По умолчанию генерируется описание элемента, предназначенного для за-

хвата сигналов, частота которых находится в пределах от 0 до 100 МГц.

В случае использования синхронного режима захвата контролируемых сигналов данные на выходы элементов, генерируемых на основе параметризованного модуля *Agilent Trace Core 2 (ATC2)*, могут передаваться с той же или удвоенной скоростью, по сравнению со скоростью входных данных. Требуемый вариант указывается с помощью поля выбора *TDM Rate*, которое также расположено во встроенной панели *State Options*. По умолчанию для параметра *TDM Rate* предлагается значение 1x, при котором данные на выходы генерируемого элемента транслируются с той же скоростью, что и поступают на входы. Чтобы установить удвоенную скорость передачи данных на выходы формируемого компонента, в поле выбора *TDM Rate* следует указать значение 2x.

Вторая диалоговая панель «мастера» настройки параметризованного модуля *Agilent Trace Core 2 (ATC2)* версии v1.01a используется для определения значений параметров выходов генерируемого элемента и количества банков входных сигналов. Вид этой диалоговой панели показан на рис. 18.

Индикатор состояния *Enable Auto Setup*, находящийся во встроенной панели *Pin Settings*, предоставляет возможность использования режима автоматической настройки выходных портов в генерируемом элементе. По умолчанию данный индикатор находится в состоянии «Включено», разрешающем применение указанного режима. В этом случае логический анализатор компании Agilent Technologies осуществляет автоматическую коммутацию входных контактов и соответствующих выходов элемента, формируемого с помощью параметризованного модуля *Agilent Trace Core 2 (ATC2)*. При этом также производится автоматический выбор уровней сигналов на каждом выходе создаваемого элемента. Для отключения режима автоматической настройки параметров вы-

ходных портов следует переключить индикатор *Enable Auto Setup* в сброшенное состояние.

Для осуществления контроля состояния внутренних сигналов разрабатываемого устройства или встраиваемой микропроцессорной системы сразу же после конфигурирования ПЛИС следует воспользоваться индикатором состояния *Enable Always On Mode*, который также расположен во встроенной панели *Pin Settings*. По умолчанию для данного параметра предлагается значение «Выключено», блокирующее возможность автоматического захвата выбранных сигналов сразу же после завершения процесса перевода кристалла в активное состояние. При переключении индикатора *Enable Always On Mode* в состояние «Включено» генерируемый элемент позволяет выполнять автоматический захват контролируемых сигналов непосредственно после завершения конфигурирования ПЛИС. Параметр *Enable Always On Mode* можно использовать только при выборе асинхронного режима захвата *Timing — Asynchronous Sampling* в стартовой диалоговой панели.

Количество используемых выходных контактов в генерируемом элементе, подключаемых к входам логического анализатора компании Agilent Technologies, указывается в поле редактирования *ATD Pin Count* (рис. 18). Допустимый диапазон изменения значений этого параметра отображается в строке *Range*, расположенной справа от поля редактирования *ATD Pin Count*.

Тип выходов данных и синхронизации создаваемого элемента указывается в поле выбора *Driver Endpoint Type*, которое находится во встроенной панели *Pin Settings*. Выпадающий список возможных значений этого поля выбора содержит два варианта: *Single-Ended* и *Differential*. При использовании варианта *Single-Ended*, предлагаемого по умолчанию, в генерируемом элементе формируются обычные (однополюсные) выходы дан-

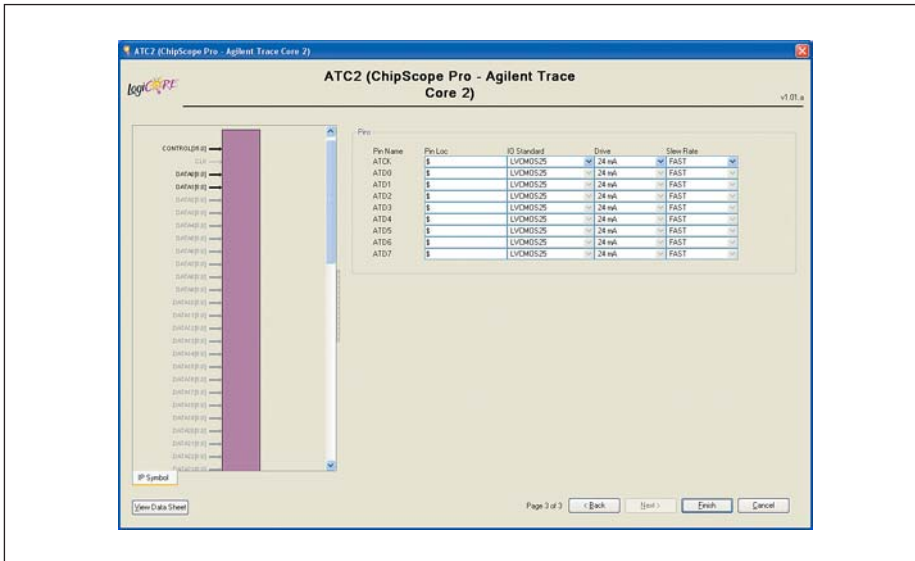


Рис. 19. Вид третьей диалоговой панели «мастера» настройки параметров ядра Agilent Trace Core 2 (ATC2) версии v1.01a

ных и синхронизации. Чтобы использовать дифференциальный тип выходов, следует в поле выбора *Driver Endpoint Type* указать вариант *Differential*.

Параметризованный модуль *Agilent Trace Core 2 (ATC2)* версии v1.01a позволяет устанавливать одинаковые значения основных параметров (стандарта ввода/вывода, нагрузочной способности и скорости переключения фронтов) для всех выходов (выходов данных и синхронизации) в генерируемом элементе или индивидуально для каждого выхода в отдельности. Выбор требуемого варианта установки значений параметров осуществляется с помощью двух кнопок с зависимой фиксацией *ATD drivers same as ATCK* и *ATD drivers different than ATCK*. Чтобы использовать одинаковые значения параметров для всех выходов формируемого элемента, следует зафиксировать в нажатом состоянии кнопку *ATD drivers same as ATCK*. При необходимости установки индивидуальных значений параметров для каждого выхода нужно переключить в нажатое положение кнопку *ATD drivers different than ATCK*. По умолчанию в нажатом состоянии находится кнопка *ATD drivers same as ATCK*.

Количество банков входных сигналов в элементе, формируемом на основе параметризованного модуля *Agilent Trace Core 2 (ATC2)* версии v1.01a, определяется в поле выбора *Signal Bank Count*, которое находится во встроенной панели *Signal Bank* (рис. 18). В выпадающем списке допустимых значений этого параметра представлено семь вариантов: 1, 2, 4, 8, 16, 32, 64.

Разрядность банков входных сигналов в генерируемом элементе вычисляется автоматически и отображается в строке *Signal Bank Width*, которая также представлена во встроенной панели *Signal Bank*. При выборе синхронного режима захвата контролируемых сигналов (в нажатом состоянии зафиксиро-

вана кнопка *State — Synchronous Sampling*) разрядность банков входных сигналов определяется как произведение значений параметров *ATD Pin Count* и *TDM Rate*. В случае использования асинхронного режима захвата (в нажатом положении находится кнопка *Timing — Asynchronous Sampling*) разрядность банков равна значению параметра *ATD Pin Count*, увеличенному на единицу.

Следующие диалоговые панели «мастера» настройки параметризованного модуля *Agilent Trace Core 2 (ATC2)* версии v1.01a предназначены для установки требуемых параметров выходов генерируемого элемента. Количество этих панелей зависит от значения, указанного в поле редактирования *ATD Pin Count*. Все диалоговые панели, используемые для установки значений параметров выходных портов, имеют сходную структуру. Типовой вид третьей диалоговой панели «мастера» настройки параметризованного модуля *Agilent Trace Core 2 (ATC2)* версии v1.01a представлен на рис. 19.

Каждая из этих диалоговых панелей содержит таблицу параметров выходных портов, которая имеет заголовок *Pins*. Количество доступных строк в этой таблице зависит от установленного состояния кнопок *ATD drivers same as ATCK* и *ATD drivers different than ATCK*, представленных во второй диалоговой панели «мастера» настройки параметров ядра *Agilent Trace Core 2 (ATC2)* версии v1.01a (рис. 18). Если предполагается использовать одинаковые значения параметров для всех выходов формируемого элемента (в нажатом состоянии зафиксирована кнопка *ATD drivers same as ATCK*), то в таблице *Pins* будет доступно только первая строка. Все значения параметров, указываемые в этой строке, автоматически распространяются на все выходы формируемого элемента. При выборе возможности установки индивидуальных значений параметров для каждого выхода (в на-

жатом положении находится кнопка *ATD drivers different than ATCK*) все строки данной таблицы автоматически переключаются в доступное состояние.

Таблица параметров выходных портов включает в себя пять колонок. В первой колонке с названием *Pin Name* отображается идентификатор соответствующего выхода. Вторая колонка, имеющая заголовок *Pin Loc*, предназначена для определения контактов кристалла, к которым следует подключить соответствующие выходы генерируемого элемента. Каждая ячейка в этом столбце представляет собой поле редактирования, в котором с помощью клавиатуры следует указать условное обозначение номера вывода ПЛИС. Третья колонка, имеющая название *IO Standard*, используется для выбора цифровых сигнальных стандартов, в соответствии с которыми должны конфигурироваться выходы формируемого компонента. Каждая ячейка, входящая в состав данного столбца, является полем выбора. Количество вариантов возможных значений, представленных в выпадающем списке этих полей выбора, зависит от семейства ПЛИС, используемого для реализации отлаживаемого устройства или встраиваемой микропроцессорной системы, а также от выбранного типа выходов (однополюсные или дифференциальные). В четвертом столбце рассматриваемой таблицы, имеющем заголовок *Drive*, указывается максимальная нагрузочная способность выходов создаваемого элемента. Каждая ячейка в этой колонке представляет собой поле выбора, предназначенное для определения максимального тока нагрузки, подключаемой к соответствующему выходу. Данное поле выбора находится в доступном состоянии только при использовании цифрового сигнального стандарта ввода/вывода, который поддерживает возможность установки нагрузочной способности. Выпадающие списки доступных полей выбора, представленных в колонке *Drive*, содержат значения тока нагрузки в диапазоне от 2 до 24 мА. Последняя, пятая колонка таблицы параметров выходных портов *Pins* позволяет определить скорость переключения (длительность фронтов) сигналов на соответствующих выходах генерируемого элемента. В каждой ячейке этой колонки с заголовком *Slew Rate* представлено поле выбора, выпадающий список значений которого содержит два варианта: *FAST* и *SLOW*. Значение *Fast*, предлагаемое по умолчанию, соответствует режиму быстрого переключения выходных сигналов. При выборе значения *Slow* снижается скорость изменения (увеличиваются фронты) выходных сигналов, что позволяет снизить уровень паразитных колебаний (шума) на выходах формируемого элемента.

Последовательно открывая диалоговые панели, вид которых соответствует изображению, представленному на рис. 19, следует задать необходимые значения параметров для

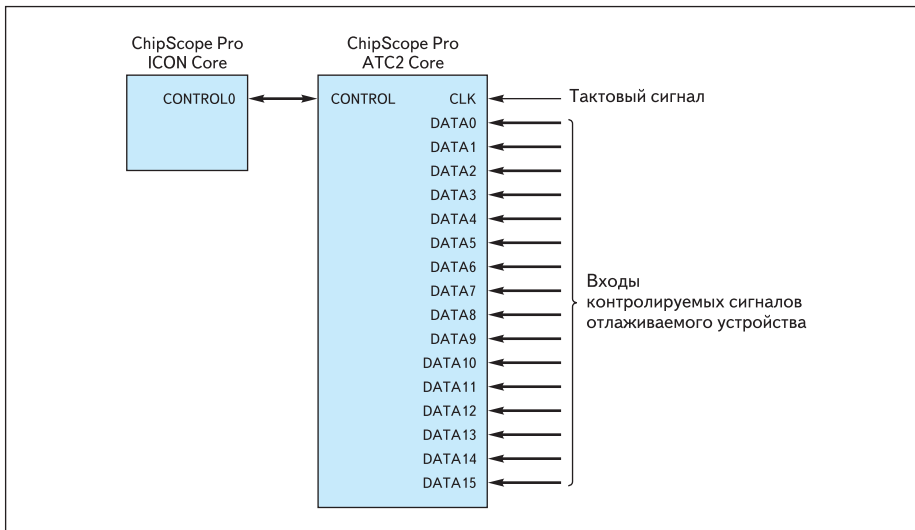


Рис. 20. Схема сопряжения элемента, формируемого на основе параметризованного модуля Agilent Trace Core 2 (ATC2) версии v1.01a, с контроллером ICON и цепями отлаживаемого устройства

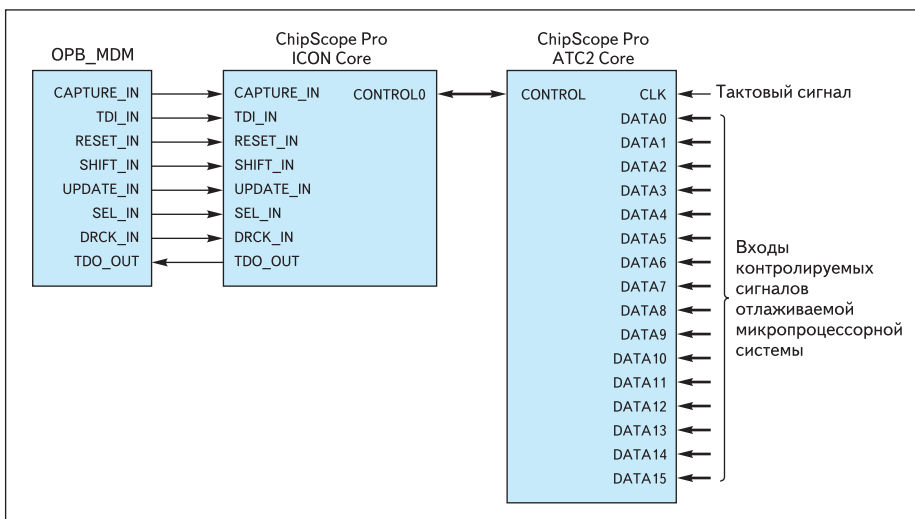


Рис. 21. Схема включения элемента, формируемого на основе параметризованного модуля Agilent Trace Core 2 (ATC2) версии v1.01a, в состав отлаживаемых встраиваемых микропроцессорных систем, разрабатываемых на базе 32-разрядных ядер семейств PowerPC и MicroBlaze

каждого выходного порта создаваемого элемента. После завершения процедуры определения значений параметров запускается процесс автоматической генерации описания ядра, предоставляющего возможность использования аппаратных логических анализаторов компанией Agilent Technologies в процессе внутрикристалльной отладки разрабатываемых устройств и встраиваемых микропроцессорных систем.

При успешном окончании этого процесса нужно в рамках проекта, разрабатываемого в САПР серии Xilinx ISE [3, 4], добавить в состав описания проектируемого устройства экземпляр данного элемента и подключить соответствующие входы и выходы этого компонента к контроллеру ICON и требуемым цепям отлаживаемого устройства в соответствии со схемой, представленной на рис. 20. Неиспользуемые в процессе внутрикристал-

ной отладки входные синхронные и асинхронные порты должны быть подключены к цепи низкого логического уровня сигнала. Это необходимо для того, чтобы избежать предупреждений о наличии неподсоединенных входных цепей и удаления триггеров и регистров с неподключенными входами в процессе оптимизации, осуществляемой средствами размещения и трассировки проекта в кристалле.

В элементах, создаваемых на основе параметризованного модуля Agilent Trace Core 2 (ATC2) версии v1.01a, применяются следующие условные обозначения входных и выходных портов:

- CLK — вход сигнала синхронизации отлаживаемого устройства или встраиваемой микропроцессорной системы;
- CONTROL[35:0] — 36-разрядная двунаправленная шина, предназначенная для со-

пряжения формируемых элементов с контроллером Integrated Controller (ICON);

- DATAn[M:0] — входная шина данных с разрядностью M+1, предназначенная для подключения n-й группы (банка) регистрируемых сигналов отлаживаемого устройства или встраиваемой микропроцессорной системы.

Количество входных шин данных DATAn[M:0], присутствующих в составе интерфейса генерируемого элемента, определяется общим числом банков регистрируемых сигналов, выбранным в процессе установки параметров ядра. Выходы данных ATDm[K:0] и синхронизации ATСК не включаются в состав интерфейса формируемых элементов. Эти выходные порты подключаются непосредственно к выводам кристалла, которые были указаны в таблице Pins, представленной в третьей диалоговой панели «мастера» настройки параметров ядра Agilent Trace Core 2 (ATC2) версии v1.01a (рис. 19).

Элементы, создаваемые с помощью параметризованного модуля Agilent Trace Core 2 (ATC2) версии v1.01a, предоставляют возможность применения аппаратных логических анализаторов, выпускаемых компанией Agilent Technologies, в процессе внутрикристалльной отладки встраиваемых систем, проектируемых на основе 32-разрядных микропроцессорных ядер семейств PowerPC и MicroBlaze [6–15]. Для этой цели необходимо включить сгенерированное ядро в состав проекта, разрабатываемого в среде комплекса Xilinx EDK [5]. Схема включения элемента, формируемого на основе параметризованного модуля Agilent Trace Core 2 (ATC2) версии v1.01a, в состав отлаживаемых встраиваемых микропроцессорных систем, разрабатываемых на базе 32-разрядных ядер семейств PowerPC и MicroBlaze, приведена на рис. 21.

Пример описания элемента, сформированного на основе параметризованного модуля Agilent Trace Core 2 (ATC2) версии v1.01a с помощью средств CORE Generator

Результат применения параметризованного модуля Agilent Trace Core 2 (ATC2) версии v1.01a показан в настоящем разделе на примере VHDL-описания элемента *chipscope_atc2_v1_01_a*. Этот элемент предоставляет возможность осуществления контроля двух групп (банков) сигналов с помощью аппаратного логического анализатора компании Agilent Technologies с использованием асинхронного режима захвата. Каждый банк позволяет осуществлять мониторинг состояния пяти входных сигналов. Сформированное ядро реализуется оптимальным образом с учетом диапазона контролируемых сигналов от 0 до 100 МГц. В элементе *chipscope_atc2_v1_01_a* задействован режим автоматической настройки параметров выходных портов. В дан-

ном ядре предусмотрена возможность автоматического захвата контролируемых сигналов непосредственно после завершения конфигурирования кристалла, используемого для реализации отлаживаемого устройства или встраиваемой микропроцессорной системы. Все выходы данных сгенерированного элемента имеют одинаковые параметры и обладают совместимостью по уровням сигналов со стандартом LVCMOS25. При этом обеспечивается поддержка тока нагрузки до 24 мА и режима быстрого переключения выходных сигналов.

Сформированный текст VHDL-описания элемента *chipscope_atc2_v1_01_a* имеет ту же структуру, что и описания других компонентов, формируемых с помощью параметризованных модулей *Xilinx CORE Generator Tool*. Первая часть содержит ссылки на используемые стандартные библиотеки IEEE, унифицированные библиотеки UNISIM фирмы Xilinx и пакеты этих библиотек, а также объявление объекта описания:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VCOMPONENTS.ALL;
use UNISIM.VPKG.ALL;
--
entity chipscope_atc2_v1_01_a is
port (
CONTROL : inout STD_LOGIC_VECTOR ( 35 downto 0 );
DATA0 : in STD_LOGIC_VECTOR ( 4 downto 0 );
DATA1 : in STD_LOGIC_VECTOR ( 4 downto 0 );
);
end chipscope_atc2_v1_01_a;
```

Во второй части представлены выражения декларации внутренних сигналов, которые используются в структурном описании архитектуры сгенерированного элемента *chipscope_atc2_v1_01_a*:

```
architecture STRUCTURE of chipscope_atc2_v1_01_a is
component OPAD
port ( OPAD : in STD_LOGIC := 'X' );
end component;
signal N0 : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_s_rst_i : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_s_sel_i : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datard_ld : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datard_msel : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datawr_cnt_7 : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datawr_cnt_rst : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datawr_msel : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_cfg_acb_datard_edge : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_cfg_acb_datawr_inv : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_sstat_o : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_tdo_next : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_addrctl_sh_ce : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_datard_sh_ce : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_sh_ce : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_rd_edge_d_edge_0 : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_rd_edge_gate_in_inv : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr_rst : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_0_Q : STD_LOGIC;
...
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_Q : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_sbm_sel : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_I_DGEN_A_u_rcd_x11_reg_ce : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_I_DGEN_A_u_rsd_x11_reg_rd_ce : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_I_MBG1T_u_rcd_x12_reg_ce : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_I_X13_DGEN_u_rcd_x13_reg_wr : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x10_reg_ce : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x12 : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_reg_ad_x10 : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_reg_ad_x10_nib0 : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_reg_ad_x11 : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_reg_ad_x11_nib0 : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_reg_ad_x12 : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_reg_ad_x12_nib0 : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_reg_ad_x13 : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_reg_ad_x13_nib0 : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_u_rsd_x10_reg_rd_ce : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_u_atp_iob_atp_oe_1 : STD_LOGIC;
signal NLW_U0_I_TMODEC_U_ATC_u_atp_iob_u_cpadi_dteq0_u_obuft_pad_O_UNCONNECTED : STD_LOGIC;
signal U0_I_TMODEC_U_ATC_atd : STD_LOGIC_VECTOR ( 4 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_acb_addrctl : STD_LOGIC_VECTOR ( 7 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datard : STD_LOGIC_VECTOR ( 7 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datard_d : STD_LOGIC_VECTOR ( 7 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datawr : STD_LOGIC_VECTOR ( 8 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datawr_cnt : STD_LOGIC_VECTOR ( 2 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_addrctl_d_in : STD_LOGIC_VECTOR ( 7 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_datard_d_in : STD_LOGIC_VECTOR ( 7 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_d_in : STD_LOGIC_VECTOR ( 8 downto 0 );
```

```
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_dwr_cnt_I_LDF_u_count_CI : STD_LOGIC_VECTOR ( 2 downto 1 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_dwr_cnt_I_LDF_u_count_D : STD_LOGIC_VECTOR ( 2 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_dwr_cnt_I_LDF_u_count_S : STD_LOGIC_VECTOR ( 2 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr : STD_LOGIC_VECTOR ( 7 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_cy : STD_LOGIC_VECTOR ( 7 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_lut : STD_LOGIC_VECTOR ( 8 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_CI : STD_LOGIC_VECTOR ( 7 downto 1 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_D : STD_LOGIC_VECTOR ( 7 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_S : STD_LOGIC_VECTOR ( 7 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_T1 : STD_LOGIC_VECTOR ( 7 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_T2 : STD_LOGIC_VECTOR ( 3 downto 0 );
signal U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_T3 : STD_LOGIC_VECTOR ( 1 downto 0 );
signal U0_I_TMODEC_U_ATC_u_atp_pwigd : STD_LOGIC_VECTOR ( 4 downto 0 );
signal U0_I_TMODEC_U_ATC_u_atp_sbm_data : STD_LOGIC_VECTOR ( 4 downto 0 );
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_muxo_x10tox11 : STD_LOGIC_VECTOR ( 7 downto 0 );
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x10 : STD_LOGIC_VECTOR ( 5 downto 0 );
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x11 : STD_LOGIC_VECTOR ( 5 downto 0 );
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_rsd_x10 : STD_LOGIC_VECTOR ( 5 downto 0 );
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_rsd_x11 : STD_LOGIC_VECTOR ( 5 downto 0 );
signal U0_I_TMODEC_U_ATC_u_atp_u_reg_sstat_data : STD_LOGIC_VECTOR ( 7 downto 0 );
signal U0_iATC_D_PAD : STD_LOGIC_VECTOR ( 4 downto 0 );
```

Третья, заключительная часть представляет собой описание архитектуры сформированного элемента, выполненное на структурном уровне. Основными компонентами, используемыми в этом описании, являются триггеры, таблицы преобразования LUT (Look-Up Table) и мультиплексоры, входящие в состав конфигурируемых логических блоков (Configurable Logic Block, CLB) ПЛИС семейств FPGA.

```
begin
XST_GND : GND
port map ( G => N0 );
XST_VCC : VCC
port map ( P => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x12 );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_rd_edge_u_edge_reg_u_fdre : FDRE
generic map ( INIT => '0' )
port map ( C => CONTROL(0),
CE => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x12,
D => CONTROL(5), R => U0_I_TMODEC_U_ATC_s_rst_i,
Q => U0_I_TMODEC_U_ATC_u_acb_bridge_u_rd_edge_d_edge_0 );
...
U0_I_TMODEC_U_ATC_u_acb_bridge_u_acb_rst_u_fdre : FDRE
generic map ( INIT => '0' )
port map ( C => CONTROL(0), CE => CONTROL(6),
D => CONTROL(1), R => N0,
Q => U0_I_TMODEC_U_ATC_s_rst_i );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_tdo : FDE
generic map ( INIT => '0' )
port map ( C => CONTROL(0),
CE => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x12,
D => U0_I_TMODEC_U_ATC_u_acb_bridge_tdo_next,
Q => CONTROL(3) );
U0_I_TMODEC_U_ATC_u_atp_iob_u_cpadi_dteq0_u_obuft_pad : OBUFT
generic map ( CAPACITANCE => «DONT_CARE»,
DRIVE => 12, IOSTANDARD => «DEFAULT» )
port map ( I => N0, T => U0_I_TMODEC_U_ATC_u_atp_iob_atp_oe_1,
O => NLW_U0_I_TMODEC_U_ATC_u_atp_iob_u_cpadi_dteq0_u_obuft_pad_O_UNCONNECTED );
...
U0_I_TMODEC_U_ATC_u_atp_iob_u_atp_oe_1 : INV
port map ( I => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x10(1),
O => U0_I_TMODEC_U_ATC_u_atp_iob_atp_oe_1 );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_datard_I_WGT1_u_ce_I1_u_or : LUT2
generic map ( INIT => X«E» )
port map ( I0 => CONTROL(5),
I1 => U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datard_ld,
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datard_sh_ce );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_datard_I_WGT1_G0_0_u_d_in_u_mux : LUT3
generic map ( INIT => X«E4» )
port map ( I0 => U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datard_ld,
I1 => U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datard(1),
I2 => U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datard(d(0)),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datard_d_in(0) );
...
U0_I_TMODEC_U_ATC_u_acb_bridge_u_dwr_cnt_I_LDF_u_count_G_0_U_XORCY : XORCY
port map ( CI => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x12,
LI => U0_I_TMODEC_U_ATC_u_acb_bridge_u_dwr_cnt_I_LDF_u_count_S(0),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_dwr_cnt_I_LDF_u_count_D(0) );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_dwr_cnt_I_LDF_u_count_G_0_GnH_U_MUXCY : MUXCY_I
port map ( CI => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x12, DI => N0,
S => U0_I_TMODEC_U_ATC_u_acb_bridge_u_dwr_cnt_I_LDF_u_count_S(0),
LO => U0_I_TMODEC_U_ATC_u_acb_bridge_u_dwr_cnt_I_LDF_u_count_CI(1) );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_dwr_cnt_I_LDF_u_count_G_0_U_LUT : LUT1
generic map ( INIT => X«2» )
port map ( I0 => U0_I_TMODEC_U_ATC_u_acb_bridge_acb_datawr_cnt(0),
```

```

O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_dwr_cnt_I_LDF_u_count_S(0);
...
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_U_rom_rd_addr_rst : INV
port map (I => CONTROL(4),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr_rst);
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_f_stat_7_I_STAT_u_stat : LUT4
generic map( INIT => X«0000» )
port map ( I0 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr(0),
I1 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr(1),
I2 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr(2),
I3 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr(3),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_7_Q );
...
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_NO_LUT6_I4_PT1_U_MUXF7 : MUXF7
port map ( I0 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_T3(0),
I1 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_T3(1),
S => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr(4),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_sstat_o );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_NO_LUT6_I4_FL_1_U_MUXF6 : MUXF6
port map ( I0 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_T2(1),
I1 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_T2(3),
S => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr(5),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_T3(1) );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_NO_LUT6_I4_FK_2_U_MUXF5 : MUXF5
port map ( I0 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_T1(2),
I1 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_T1(6),
S => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr(6),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_T2(2) );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_NO_LUT6_I4_FJ_7_U_LUT3 : LUT3
generic map( INIT => X«CA» )
port map ( I0 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_7_Q,
I1 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_Q,
I2 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr(7),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_smux_T1(7) );
...
U0_I_TMODEC_U_ATC_u_acb_bridge_u_dwr_cnt_rst_I1_u_or : LUT2
generic map( INIT => X«E» )
port map ( I0 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_srst_i,
I1 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_inv,
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_cnt_rst );
...
U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_inv_inv : INV
port map ( I => CONTROL(8),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_inv );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_wrnt_7 : LUT3
generic map( INIT => X«80» )
port map ( I0 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_cnt(0),
I1 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_cnt(1),
I2 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_cnt(2),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_cnt_7 );
...
U0_G_DATAPADS_3_I_DATAPAD : OPAD
port map ( OPAD => U0_iATC_D_PAD(3) );
...
U0_G_DATAPADS_0_I_DATAPAD : OPAD
port map ( OPAD => U0_iATC_D_PAD(0) );
U0_I_TMODEP_I_CLKPAD : OPAD
port map ( OPAD => U0_iATC_D_PAD(4) );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_G_0_U_XORCY : XORCY
port map ( CI => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x12,
I1 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_S(0),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_D(0) );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_G_0_GnH_U_MUXCY : MUXCY_L
port map ( CI => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x12, DI => N0,
S => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_S(0),
LO => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_CI(1) );
...
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_G_0_U_LUT : LUT1
generic map( INIT => X«2» )
port map ( I0 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr(0),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_S(0) );
...
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_G_0_U_FDRE : FDRE
generic map( INIT => '0' )
port map ( C => CONTROL(0), CE => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x12,
D => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_u_rom_rd_addr_D(0),
R => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr_rst,
Q => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_rd_addr(0) );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_I_WGT1_G0_8_u_d_in_u_mux : LUT3
generic map( INIT => X«E4» )
port map ( I0 => N0, I1 => CONTROL(1), I2 => N0,
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_d_in(8) );
...
U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_I_WGT1_u_ce_I1_u_or : LUT2
generic map( INIT => X«E» )
port map ( I0 => CONTROL(8), I1 => N0,
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_sh_ce );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_I_WGT1_G0_8_u_reg_u_fdre : FDRE
generic map( INIT => '0' )
port map ( C => CONTROL(0),
CE => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_sh_ce,
D => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr_d_in(8),
R => U0_I_TMODEC_U_ATC_u_srst_i,
Q => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr(8) );
...
U0_I_TMODEC_U_ATC_u_atp_u_reg_u_sstat_u_sstat_G0_0_u_rom_u_lut4 : LUT4
generic map( INIT => X«901D» )
port map ( I0 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_acb_addrctl(0),

```

```

I1 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_acb_addrctl(1),
I2 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_acb_addrctl(2),
I3 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_acb_addrctl(3),
O => U0_I_TMODEC_U_ATC_u_atp_u_reg_u_mux_x10tox11_G0_7_u_mux_NO_LUT6_I1_U_LUT3 : LUT3
generic map( INIT => X«CA» )
port map ( I0 => N0, I1 => N0,
I2 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_acb_addrctl(0),
O => U0_I_TMODEC_U_ATC_u_atp_u_reg_u_mux_x10tox11(7) );
U0_I_TMODEC_U_ATC_u_atp_u_reg_I_X13_DGEN_u_reg_ad_x13 : LUT2
generic map( INIT => X«8» )
port map ( I0 => U0_I_TMODEC_U_ATC_u_atp_u_reg_reg_ad_x13_nib0,
I1 => U0_I_TMODEC_U_ATC_u_acb_bridge_u_acb_addrctl(4),
O => U0_I_TMODEC_U_ATC_u_atp_u_reg_reg_ad_x13 );
...
U0_I_TMODEC_U_ATC_u_atp_u_reg_I_X13_DGEN_u_rcd_x13_g_wdeql1_u_reg_G0_0_u_reg_u_fdre : FDRE
generic map( INIT => '0' )
port map ( C => CONTROL(0),
CE => U0_I_TMODEC_U_ATC_u_atp_u_reg_I_X13_DGEN_u_rcd_x13_reg_wr,
D => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr(1),
R => U0_I_TMODEC_U_ATC_u_srst_i,
Q => U0_I_TMODEC_U_ATC_u_atp_u_reg_pwigd(0) );
...
U0_I_TMODEC_U_ATC_u_atp_u_reg_I_ALWAYS_ON_u_rcd_x10_fdse_1 : FDSE
generic map( INIT => '1' )
port map ( C => CONTROL(0),
CE => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x10_reg_ce,
D => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr(2),
S => U0_I_TMODEC_U_ATC_u_srst_i,
Q => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x10(1) );
U0_I_TMODEC_U_ATC_u_atp_u_reg_u_rcd_x10_fdre_0 : FDRE
generic map( INIT => '0' )
port map ( C => CONTROL(0),
CE => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x10_reg_ce,
D => U0_I_TMODEC_U_ATC_u_acb_bridge_u_datawr(1),
R => U0_I_TMODEC_U_ATC_u_srst_i,
Q => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x10(0) );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_lut_0_Q : LUT3
generic map( INIT => X«800» )
port map ( I0 => CONTROL(14), I1 => CONTROL(11), I2 => CONTROL(12),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_lut(0) );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_cy_0_Q : MUXCY
port map ( CI => N0, DI => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x12,
S => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_lut(0),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_cy(0) );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_lut_1_Q : LUT4
generic map( INIT => X«8000» )
port map ( I0 => CONTROL(13), I1 => CONTROL(15),
I2 => CONTROL(10), I3 => CONTROL(16),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_lut(1) );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_lut_8_Q : LUT4
generic map( INIT => X«8000» )
port map ( I0 => CONTROL(35), I1 => CONTROL(0),
I2 => CONTROL(5), I3 => CONTROL(1),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_lut(8) );
U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_cy_8_Q : MUXCY
port map (
CI => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_cy(7),
DI => U0_I_TMODEC_U_ATC_u_atp_u_reg_rcd_x12,
S => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_wg_lut(8),
O => U0_I_TMODEC_U_ATC_u_acb_bridge_u_static_stat_rom_sstat_o_10_Q );
end STRUCTURE;

```

Для декларации сформированного компонента *chipscope_atc2_v1_01_a* нужно добавить в соответствующий раздел исходного VHDL-описания следующую конструкцию:

```

component chipscope_atc2_v1_01_a
PORT (
CONTROL : INOUT STD_LOGIC_VECTOR(35 DOWNTO 0);
DATA0 : IN STD_LOGIC_VECTOR(4 DOWNTO 0);
DATA1 : IN STD_LOGIC_VECTOR(4 DOWNTO 0)
);
end component;

```

Конкретный экземпляр сгенерированного компонента, обеспечивающего возможность применения аппаратных логических анализаторов компании Agilent Technologies в процессе внутрикристалльной аппаратной отладки разрабатываемого устройства, создается с помощью оператора, шаблон которого выглядит следующим образом:

```

<идентификатор_экземпляра_элемента_chipscope_atc2_v1_01_a>: chipscope_atc2_v1_01_a
port map (
CONTROL => CONTROL,
DATA0 => DATA0,
DATA1 => DATA1
);

```

Детализированные сведения о количестве различных ресурсов кристалла (триггеров Flip-Flop, таблиц преобразования LUT и секций

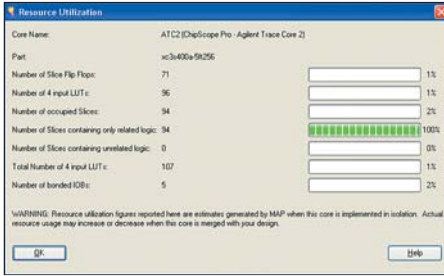


Рис. 22. Вид информационной панели, содержащей сведения об объеме различных ресурсов кристалла, используемых для автономной реализации элемента `chipscope_atc2_v1_01_a`

Slices), необходимых для автономной реализации элемента `chipscope_atc2_v1_01_a`, содержатся в информационной панели, вид которой показан на рис. 22.

Заключение

После генерации описаний всех необходимых отладочных элементов с помощью параметризованных модулей Xilinx CORE Generator Tool, рассмотренных в настоящей статье, следует включить требуемое количество экземпляров этих компонентов в состав структуры разрабатываемого устройства или встраиваемой микропроцессорной системы. При этом соответствующие входы и выходы этих компонентов нужно подключить к цепям тех сигналов, состояние которых предполагается контролировать или изменять в процессе отладки. Затем необходимо выполнить этапы синтеза, размещения и трассировки в кристалле, а также создания конфигурационной последовательности проектируемого устройства или микропроцессорной системы. Выполнив загрузку сгенерированной конфигурационной последовательности в кристалл, можно приступить к аппаратной отладке разработан-

ного устройства или встраиваемой системы, используя для этого программные средства ChipScope Pro Analyzer [1] или логический анализатор, выпускаемый компанией Agilent Technologies. Работа с программными средствами ChipScope Pro Analyzer будет рассмотрена в одной из последующих публикаций цикла, посвященного вопросам аппаратной внутрикристалльной отладки цифровых устройств и встраиваемых микропроцессорных систем, проектируемых на базе ПЛИС фирмы Xilinx с архитектурой FPGA.

Литература

1. Зотов В. Средства внутрикристалльной отладки цифровых устройств и встраиваемых микропроцессорных систем, разрабатываемых на базе ПЛИС с архитектурой FPGA фирмы Xilinx — ChipScope Pro // Компоненты и технологии. 2008. № 10.
2. Кузелин М. О., Кнышев Д. А., Зотов В. Ю. Современные семейства ПЛИС фирмы Xilinx: Справочное пособие. М.: Горячая линия — Телеком, 2004.
3. Зотов В. Проектирование цифровых устройств, реализуемых на базе ПЛИС FPGA фирмы Xilinx, с использованием средств CORE Generator // Компоненты и технологии. 2006. № 12. 2007. № 1.
4. Зотов В. Проектирование цифровых устройств на основе ПЛИС фирмы Xilinx в САПР WebPack ISE. М.: Горячая линия — Телеком, 2003.
5. Зотов В. Embedded Development Kit — система проектирования встраиваемых микропроцессорных систем на основе ПЛИС серий FPGA фирмы Xilinx // Компоненты и технологии. 2004. № 4.
6. Зотов В. Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы Xilinx. М.: Горячая линия — Телеком, 2006.
7. Зотов В. MicroBlaze — семейство 32-разрядных микропроцессорных ядер, реализуемых на основе ПЛИС фирмы Xilinx // Компоненты и технологии. 2003. № 9.

8. Зотов В. Система команд микропроцессорного ядра MicroBlaze // Компоненты и технологии. 2004. № 1–3.
9. Зотов В. Организация памяти микропроцессорного ядра MicroBlaze // Компоненты и технологии. 2004. № 5.
10. Зотов В. Создание проекта микропроцессорной системы на основе ядра MicroBlaze, реализуемой в ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2004. № 6.
11. Зотов В. Формирование спецификации аппаратной платформы микропроцессорной системы на основе ядра MicroBlaze, реализуемой в ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2004. № 7–8.
12. Зотов В. Реализация аппаратной платформы микропроцессорной системы, проектируемой на основе ядра MicroBlaze, в ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2004. № 9.
13. Зотов В. Разработка спецификации программных средств микропроцессорной системы на основе ядра MicroBlaze, реализуемой в ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2005. № 1.
14. Зотов В. Разработка программного обеспечения микропроцессорной системы на основе ядра MicroBlaze, реализуемой в ПЛИС семейств FPGA фирмы Xilinx // Компоненты и технологии. 2005. № 2.
15. Зотов В. Разработка встраиваемых микропроцессорных систем на основе ядра MicroBlaze, реализуемых в ПЛИС семейств FPGA фирмы Xilinx, с помощью «мастера» Base System Builder Wizard // Компоненты и технологии. 2005. № 3–4.
16. Зотов В. Новый инструментальный комплект от компании Avnet на основе ПЛИС FPGA семейства Spartan-3A фирмы Xilinx // Компоненты и технологии. 2008. № 8.
17. Зотов В. Инструментальный модуль компании Avnet для отладки проектов встраиваемых систем, разрабатываемых на базе нового семейства ПЛИС FPGA фирмы Xilinx Virtex -5 FXT // Компоненты и технологии. 2008. № 9.