Формирование описаний компонентов для внутрикристальной отладки цифровых устройств и встраиваемых микропроцессорных систем

на основе параметризированных модулей Xilinx CORE Generator Tool

Валерий ЗОТОВ
walerry@km.ru

В предыдущей публикации [1] цикла, посвященного вопросам аппаратной внутрикристальной отладки цифровых устройств и встраиваемых микропроцессорных систем, проектируемых на базе ПЛИС фирмы Xilinx с архитектурой FPGA (Field Programmable Gate Array) [2], были представлены функциональные возможности и структура комплекса программных средств ChipScope Pro, а также методы его применения.

Для подключения отладочных компонентов к соответствующим цепям разрабатываемого устройства на этапе подготовки модулей исходного описания проекта необходимо предварительно сгенерировать соответствующие элементы на основе параметризированных модулей Xilinx CORE Generator Tool. В настоящей статье рассматривается процесс формирования описаний компонентов для внутрикристальной отладки цифровых устройств и встраиваемых микропроцессорных систем, выполняемый с помощью генератора параметризированных модулей Xilinx CORE Generator [3].

Подготовку описаний отладочных элементов можно осуществлять как в автономном режиме работы средств Xilinx CORE Generator, так и при их запуске в среде управляющей оболочки САПР серии Xilinx ISE (Integrated Synthesis Environment/Integrated Software Environment) [4] *Навигатора проекта* (*Project Navigator*). После активизации генератора параметризированных модулей в автономном режиме необходимо создать новый проект, указав язык HDL (Hardware Description Language), используемый для описания формируемых элементов, семейство ПЛИС и конкретный тип кристалла, для которого создаются отладочные элементы. Эта процедура подробно рассмотрена в [3]. Далее, независимо от способа запуска средств Xilinx CORE Generator, следует на странице *View by Function* встроенного окна выбора IP-ядер открыть папку *Debug & Verification*, которая содержит параметризированные модули, предназначенные для аппаратной отладки разрабатываемых устройств. В этой папке нужно открыть раздел *ChipScope Pro* и поочередно выбрать ядра, входящие в состав набора *Xilinx CORE Generator Tool*, для генерации требуемых отладочных компонентов, как показано на рис. 1. Выбор конфигурации формируемых компонентов производится в диалоговом режиме с помощью «мастера» настройки параметров соответствующего ядра.

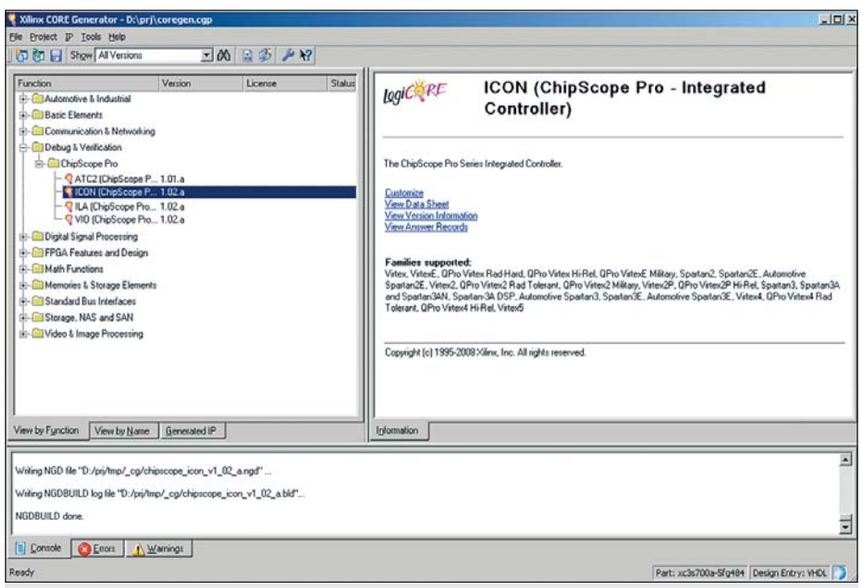

Рис. 1. Вид страницы View by Function

Генерация описания контроллера, осуществляющего взаимодействие конфигурируемых логических анализаторов и виртуальных входов/выходов с портом JTAG-интерфейса, на основе параметризированного модуля Integrated Controller (ICON) с помощью средств CORE Generator

Обязательным элементом, необходимым для организации процесса аппаратной внутрикристальной отладки разрабатываемых устройств и встраиваемых микропроцессорных систем, является контроллер, осуществляющий взаимодействие конфигурируемых логических анализаторов *Integrated Logic Analyzer* (*ILA*) и виртуальных входов/выходов *Virtual Input/Output* (*VIO*), а также конфигурируемых анализаторов шин *Integrated Bus Analyzer* (*IBA*), соответствующих спецификации CoreConnect OPB (On-Chip Peripheral Bus) с портом JTAG-интерфейса. Для подготовки HDL-описания данного контроллера в составе генератора параметризированных модулей CORE Generator предусмотрено ядро *Integrated Controller* (*ICON*). Актуальной версией этого ядра к моменту подготовки настоящей статьи являлась модификация Integrated Controller (ICON) v1.02a, которая позволяет формировать описания контроллеров, предназначенных для аппаратной отладки проектов, реализуемых на основе ПЛИС следующих семейств: Spartan-II, Spartan-IIE, Spartan-3, Spartan-3 XA, Spartan-3E, Spartan-3E XA, Spartan-3A, Spartan-3AN, Spartan-3A DSP, Virtex QPRO Virtex Rad-Hard, QPRO Virtex Hi-Rel, Virtex-E, QPRO Virtex-E Military, 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, Virtex-5 FXT.

Основные отличия параметризированного модуля *Integrated Controller* (*ICON*) версии v1.02a следующие:

- возможность выбора количества портов, предназначенных для подключения конфигурируемых логических анализаторов, виртуальных входов/выходов и конфигурируемых анализаторов шин, соответствующих спецификации CoreConnect OPB, в диапазоне от 1 до 15;
- поддержка различных методов использования элементов BSCAN в генерируемых контроллерах;
- возможность применения создаваемых контроллеров для отладки встраиваемых микропроцессорных систем, разрабатываемых с помощью комплекса средств проектирования Xilinx Embedded Development Kit (EDK) [5, 6].

Выбор конфигурации контроллера, формируемого на основе рассматриваемого параметризированного модуля, осуществляется с помощью соответствующего «мастера» настройки параметров, который содержит одну диалоговую панель, вид которой представлен на рис. 2.

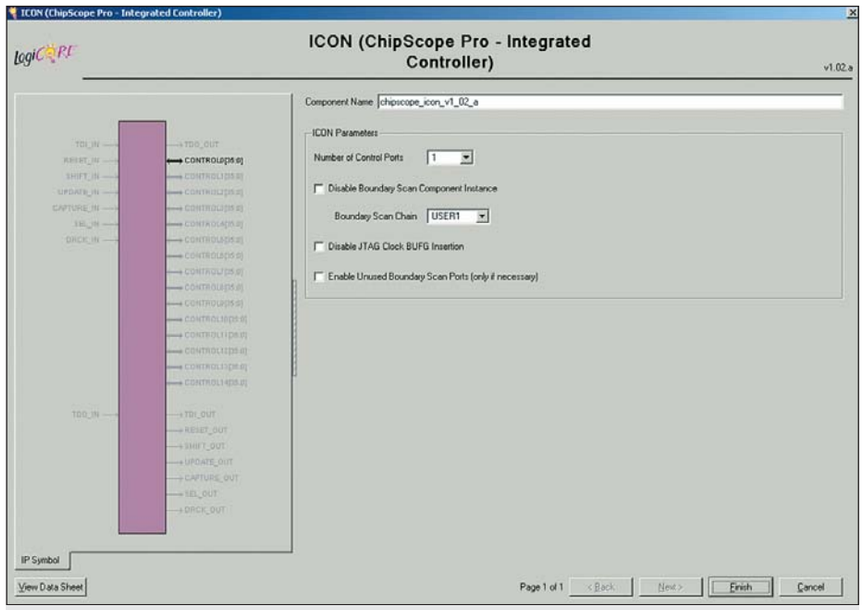

Рис. 2. Вид диалоговой панели «мастера» настройки параметров ядра контроллера Integrated Controller (ICON) версии v1.02a

В этой диалоговой панели, прежде всего, нужно указать название создаваемого контроллера в поле редактирования *Component Name*. Первоначально при выводе данной диалоговой панели в поле *Component Name* отображается идентификатор, предлагаемый по умолчанию. Для редактирования данного названия или ввода нового идентификатора следует воспользоваться клавиатурой. При этом необходимо учитывать следующие правила, которые относятся ко всем параметризированным модулям, входящим в состав набора *Xilinx CORE Generator Tool*. Во-первых, в составе названия формируемого ядра (контроллера, конфигурируемого логического анализатора или виртуального входа/выхода) можно использовать только строчные буквы латинского алфавита (a–z), цифры (0–9) и символ подчеркивания (_). Во-вторых, указываемый идентификатор должен начинаться со строчной буквы. В-третьих, в качестве названий генерируемых элементов нельзя использовать ключевые слова, зарезервированные в языках описания аппаратуры VHDL и Verilog, как, например, component, port и т. п.

Далее следует определить количество портов в формируемом контроллере, которые предназначены для подключения других отладочных компонентов (конфигурируемых логических анализаторов, виртуальных входов/выходов, конфигурируемых анализаторов шин). Для этой цели нужно воспользоваться полем выбора *Number of Control Ports*, которое расположено во встроенной панели *ICON Parameters* (рис. 2). Выпадающий список этого поля выбора содержит значения в диапазоне от 1 до 15. По умолчанию для параметра *Number of Control Ports* предлагается единичное значение. При выборе иного значения данного параметра следует учитывать, что количество портов в генерируемом контроллере должно соответствовать общему числу конфигурируемых логических анализаторов, виртуальных входов/выходов и анализаторов шин, используемых в процессе отладки разрабатываемого устройства или встраиваемой микропроцессорной системы. Не рекомендуется оставлять порты контроллера, формируемого на основе параметризированного модуля *Integrated Controller* (*ICON*), в неподключенном состоянии. При обнаружении неподключенного порта контроллера *ICON* средствами размещения и трассировки проекта в кристалле ПЛИС выводится сообщение об ошибке.

Если для реализации отлаживаемого устройства выбран кристалл семейства Spartan-II, Spartan-IIE, Spartan-3, Spartan-3 XA, Spartan-3E, Spartan-3E XA, Spartan-3A, Spartan-3AN, Spartan-3A DSP, Virtex, QPRO Virtex Rad-Hard, QPRO Virtex Hi-Rel, Virtex-E, QPRO Virtex-E Military, Virtex-II, QPRO Virtex-II Rad Tolerant, QPRO Virtex-II Military или Virtex-II Pro, то разработчик может выбрать требуемый вариант включения элемента BSCAN. Данный компонент расширяет возможности порта Test Access Port (TAP) JTAG-интерфейса, позволяя создавать до четырех внутренних цепочек периферийного сканирования. Параметризированный модуль *Integrated Controller* (*ICON*) версии v1.02a предоставляет возможность включения примитива BSCAN в качестве компонента

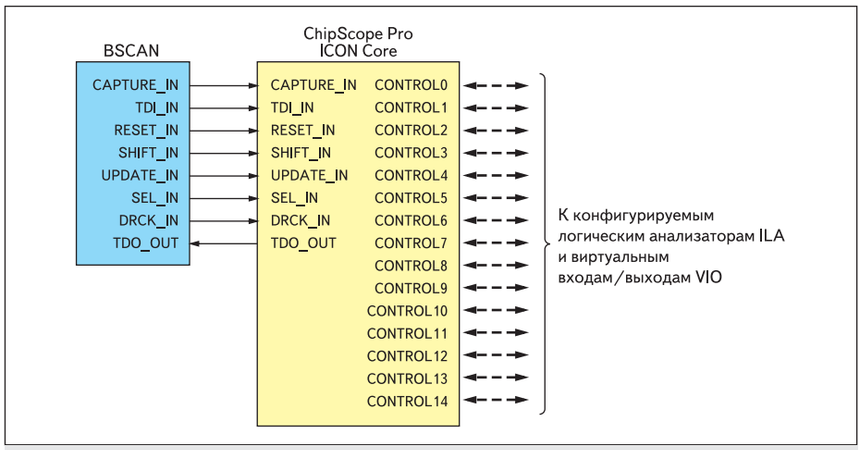

Рис. 3. Схема сопряжения контроллера ICON и внешнего (по отношению к контроллеру) элемента BSCAN

в состав формируемого описания контроллера. Выбор требуемого варианта сопряжения элемента BSCAN и контроллера *ICON* осуществляется с помощью индикатора состояния *Disable Boundary Scan Component Instance*. По умолчанию этот индикатор установлен в состояние «Выключено», при котором элемент BSCAN, соответствующий выбранному семейству ПЛИС, включается в состав формируемого контроллера. Для того чтобы сгенерировать описание контроллера, не содержащего указанного компонента, следует перевести индикатор *Disable Boundary Scan Component Instance* в состояние «Включено». При этом взаимодействие контроллера и внешнего (по отношению к контроллеру) элемента BSCAN осуществляется в соответствии с функциональной схемой, представленной на рис. 3.

В случае применения генерируемого контроллера *ICON* для отладки встраиваемых систем, разрабатываемых на базе 32-разрядных микропроцессорных ядер семейств PowerPC и MicroBlaze [6–15] в рамках комплекса средств проектирования Xilinx EDK, его подключение осуществляется с помощью периферийного компонента OPB_MDM, как показано на рис. 4.

Следующим шагом в процессе подготовки описания контроллера *ICON*, в состав которого включен компонент BSCAN, является выбор номера (идентификатора) цепочки периферийного сканирования, к ней должен быть предоставлен доступ программным средствам *ChipScope Pro Analyzer*. Данные средства используются для управления процессом отладки разрабатываемого устройства и отображения результатов этого процесса на экране монитора. Для определения номера используемой цепочки периферийного сканирования предназначено поле выбора *Boundary Scan Chain*, которое также находится во встроенной панели *ICON Parameters* (рис. 2). В выпадающем списке этого поля выбора представлены идентификаторы цепочек периферийного сканирования, соответствующие выбранному семейству ПЛИС (USER1, USER2, USER3 и USER4). По умолчанию поле выбора *Boundary Scan Chain* содержит вариант USER1.

При включении элемента BSCAN в состав формируемого описания контроллера *ICON*

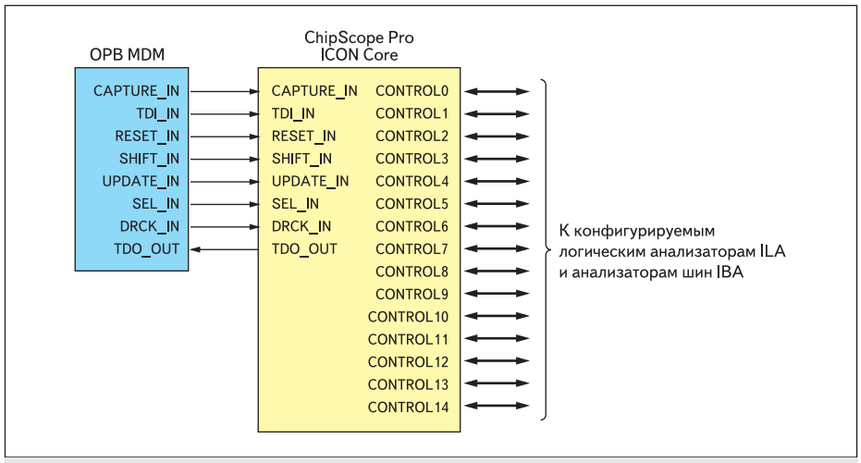

Рис. 4. Схема включения контроллера ICON в состав отлаживаемой встраиваемой микропроцессорной системы

в параметризированном модуле *Integrated Controller* (*ICON*) версии v1.02a предусмотрена возможность выбора типа буферного элемента, устанавливаемого в цепи тактового сигнала JTAG-интерфейса, и, соответственно, типа ресурсов, используемых для его последующей трассировки. Этот выбор осуществляется с помощью индикатора состояния *Disable JTAG Clock BUFG Insertion*. По умолчанию данный индикатор находится в состоянии «Выключено». При этом в цепи тактового сигнала JTAG-интерфейса применяется глобальный буферный элемент BUFG, выход которого сопряжен с глобальными цепями синхронизации ПЛИС. В тех случаях, когда для реализации проектируемого устройства необходимо максимально возможное количество глобальных буферных элементов, целесообразно установить в цепи тактового сигнала JTAG-интерфейса обычный буферный элемент. Для этого нужно перевести индикатор *Disable JTAG Clock BUFG Insertion* в состояние «Включено».

В завершение процесса определения параметров контроллера *ICON*, предназначенного для применения в ПЛИС семейств Spartan-II, Spartan-IIE, Spartan-3, Spartan-3 XA, Spartan-3E, Spartan-3E XA, Spartan-3A, Spartan-3AN, Spartan-3A DSP, Virtex, QPRO Virtex Rad-Hard, QPRO Virtex Hi-Rel, Virtex-E, QPRO Virtex-E Military, Virtex-II, QPRO Virtex-II Rad Tolerant, QPRO Virtex-II Military или Virtex-II Pro, в состав которого входит компонент BSCAN, параметризированный модуль *Integrated Controller* (*ICON*) версии v1.02a позволяет включить в состав интерфейса формируемого описания порты, открывающие доступ к неиспользуемой цепочке периферийного сканирования. Для этой цели следует воспользоваться индикатором состояния *Enable Unused Boundary Scan Ports*. Чтобы добавить в состав интерфейса генерируемого описания контроллера порты сигналов, предоставляющие возможность доступа к неиспользуемой цепочке периферийного сканирования, следует установить данный индикатор в состояние «Включено». При этом необходимо учитывать, что указанные порты должны быть обязательно задействованы в отлаживаемом проекте. Если данные порты были добавлены в состав интерфейса формируемого контроллера, но не используются в разрабатываемом устройстве (находятся в неподключенном состоянии), то в процессе синтеза такого описания, а также на этапе размещения и трассировки проекта могут возникать ошибки. По умолчанию индикатор *Enable Unused Boundary Scan Ports* находится в сброшенном состоянии, запрещающем включение портов, открывающих доступ к неиспользуемой цепочке периферийного сканирования, в состав интерфейса создаваемого контроллера *ICON*.

В параметризированном модуле *Integrated Controller* (*ICON*) используется следующая

система условных обозначений входных и выходных портов:

- CAPTURE_IN — вход сигнала CAPTURE, поступающего от внешнего элемента BSCAN;
- CAPTURE_OUT — выход сигнала CAPTURE, формируемого внутренним элементом BSCAN;
- CONTROL0[35:0] — двунаправленная шина, предназначенная для подключения первого (или единственного) конфигурируемого логического анализатора ILA или анализатора шины IBA;
- CONTROL1[35:0]–CONTROL14[35:0] — двунаправленные шины, предназначенные для подключения дополнительных конфигурируемых логических анализаторов ILA, анализаторов шин IBA и виртуальных входов/выходов VIO;
- DRCK_IN — вход сигнала DRCK, поступающего от внешнего элемента BSCAN;
- DRCK_OUT — выход сигнала DRCK, вырабатываемого внутренним элементом BSCAN;
- RESET_IN — вход сигнала RESET, поступающего от внешнего элемента BSCAN;
- RESET_OUT — выход сигнала RESET, формируемого внутренним элементом BSCAN;
- SEL_IN — вход сигнала SEL, приходящего от внешнего элемента BSCAN;
- SEL_OUT — выход сигнала SEL, вырабатываемого внутренним элементом BSCAN;
- SHIFT_IN — вход сигнала SHIFT, поступающего от внешнего элемента BSCAN;
- SHIFT_OUT — выход сигнала SHIFT, формируемого внутренним элементом BSCAN;
- TDI_IN — вход сигнала TDI, поступающего от внешнего элемента BSCAN;
- TDI_OUT — выход сигнала TDI, вырабатываемого внутренним элементом BSCAN;
- TDO_IN — вход сигнала TDO, относящегося к неиспользуемой цепочке периферийного сканирования внутреннего элемента BSCAN;
- TDO_OUT — выход сигнала TDO, предназначенного для подключения к внешнему элементу BSCAN;
- UPDATE_IN — вход сигнала UPDATE, поступающего от внешнего элемента BSCAN;
- UPDATE_OUT — выход сигнала UPDATE, вырабатываемого внутренним элементом BSCAN.

Из всех перечисленных идентификаторов в описании интерфейса сгенерированного контроллера *ICON* будут представлены условные обозначения только тех входных и выходных портов, которые соответствуют указанным пользователем значениям параметров его конфигурации.

```
architecture STRUCTURE of chipscope_icon_v1_02_a is
  signal N1 : STD_LOGIC;
  signal U0_U_ICON_U_CMD_iSEL_n : STD_LOGIC;
  signal U0_U_ICON_U_CMD_iTARGET_CE : STD_LOGIC;
  signal U0_U_ICON_U_CTRL_OUT_iDATA_VALID : STD_LOGIC;
  signal U0_U_ICON_U_STAT_iCMD_GRP0_SEL : STD_LOGIC;
  signal U0_U_ICON_U_STAT_iDATA_VALID : STD_LOGIC;
  signal U0_U_ICON_U_STAT_iSTATCMD_CE : STD_LOGIC;
  signal U0_U_ICON_U_STAT_iSTATCMD_CE_n : STD_LOGIC;
  signal U0_U_ICON_U_STAT_iSTAT_HIGH : STD_LOGIC;
  signal U0_U_ICON_U_STAT_iSTAT_LOW : STD_LOGIC;
  signal U0_U_ICON_U_STAT_iTDO_next : STD_LOGIC;
  signal U0_U_ICON_U_SYNC_iDATA_CMD_n : STD_LOGIC;
  signal U0_U_ICON_U_SYNC_iGOT_SYNC : STD_LOGIC;
  signal U0_U_ICON_U_SYNC_iGOT_SYNC_HIGH : STD_LOGIC;
  signal U0_U_ICON_U_SYNC_iGOT_SYNC_LOW : STD_LOGIC;
  signal U0_U_ICON_iCORE_ID_SEL_0_Q : STD_LOGIC;
  signal U0_U_ICON_iCORE_ID_SEL_15_Q : STD_LOGIC;
  signal U0_U_ICON_iDATA_CMD : STD_LOGIC;
  signal U0_U_ICON_iDATA_CMD_n : STD_LOGIC;
  signal U0_U_ICON_iSEL_n : STD_LOGIC;
  signal U0_U_ICON_iSYNC : STD_LOGIC;
  signal U0_U_ICON_iTDO_next : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_1_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_2_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_3_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_4_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_5_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_6_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_7_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_8_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_9_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_10_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_11_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_12_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_13_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal NLW_U0_U_ICON_U_CMD_U_CORE_ID_SEL_I4_FI_14_U_LUT_O_UNCONNECTED : STD_LOGIC;
  signal U0_U_ICON_U_CMD_iTARGET : STD_LOGIC_VECTOR ( 11 downto 8 );
  signal U0_U_ICON_U_CTRL_OUT_iCOMMAND_GRP_SEL : STD_LOGIC_VECTOR ( 1 downto 0 );
  signal U0_U_ICON_U_STAT_U_STAT_CNT_CI : STD_LOGIC_VECTOR ( 5 downto 1 );
  signal U0_U_ICON_U_STAT_U_STAT_CNT_D : STD_LOGIC_VECTOR ( 5 downto 0 );
  signal U0_U_ICON_U_STAT_U_STAT_CNT_S : STD_LOGIC_VECTOR ( 5 downto 0 );
  signal U0_U_ICON_U_STAT_iSTAT : STD_LOGIC_VECTOR ( 3 downto 0 );
  signal U0_U_ICON_U_STAT_iSTAT_CNT : STD_LOGIC_VECTOR ( 5 downto 0 );
  signal U0_U_ICON_U_SYNC_iSYNC_WORD : STD_LOGIC_VECTOR ( 6 downto 0 );
  signal U0_U_ICON_U_TDO_MUX_T1 : STD_LOGIC_VECTOR ( 7 downto 0 );
  signal U0_U_ICON_U_TDO_MUX_T2 : STD_LOGIC_VECTOR ( 3 downto 0 );
  signal U0_U_ICON_U_TDO_MUX_T3 : STD_LOGIC_VECTOR ( 1 downto 0 );
  signal U0_U_ICON_iCOMMAND_GRP : STD_LOGIC_VECTOR ( 1 downto 0 );
  signal U0_U_ICON_iCOMMAND_SEL : STD_LOGIC_VECTOR ( 15 downto 0 );
  signal U0_U_ICON_iCORE_ID : STD_LOGIC_VECTOR ( 3 downto 0 );
  signal U0_U_ICON_iTDO_VEC : STD_LOGIC_VECTOR ( 15 downto 15 );
```

Листинг 1

Пример описания контроллера, сформированного на основе параметризированного модуля Integrated Controller (ICON) с помощью средств CORE Generator

В качестве примера описания контроллера, сгенерированного на основе параметризированного модуля *Integrated Controller* (*ICON*), в настоящем разделе приводится текст VHDL-описания элемента *chipscope_icon_v1_02*. Этот контроллер предназначен для организации взаимодействия одного конфигурируемого логического анализатора ILA или анализатора шины IBA с портом JTAG-интерфейса. Контроллер *chipscope_icon_v1_02* применяется совместно с внешним элементом BSCAN. Контроллеры данного вида достаточно часто используются в процессе аппаратной внутрикристальной отладки различных устройств начального и среднего уровней сложности. Приведенное далее описание элемента *chipscope_icon_v1_02* достаточно подробно отражает внутреннюю структуру контроллера, формируемого с помощью параметризированного модуля *Integrated Controller* (*ICON*).

Для наглядности текст VHDL-описания разбит на несколько разделов. В первой части приводятся выражения декларации используемых библиотек и объявление объекта описания:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VCOMPONENTS.ALL;
use UNISIM.VPKG.ALL;
--
entity chipscope_icon_v1_02_a is
  port (
    DRCK_IN : in STD_LOGIC := 'X';
    SEL_IN : in STD_LOGIC := 'X';
    TDI_IN : in STD_LOGIC := 'X';
    UPDATE_IN : in STD_LOGIC := 'X';
    SHIFT_IN : in STD_LOGIC := 'X';
    RESET_IN : in STD_LOGIC := 'X';
    CAPTURE_IN : in STD_LOGIC := 'X';
    TDO_OUT : out STD_LOGIC;
    CONTROL0 : inout STD_LOGIC_VECTOR ( 35 downto 0 )
  );
end chipscope_icon_v1_02_a;
```

Далее следует определение архитектуры контроллера *chipscope_icon_v1_02*, выполненное на структурном уровне. В начале архитектурного тела (architecture body) представлены выражения декларации внутренних сигналов (листинг 1).

Следующая часть представляет собой структурное описание рассматриваемого контроллера. В качестве основных компонентов в этом описании используются триггеры, мультиплексоры и таблицы преобразования LUT (Look-Up Table), которые входят в состав конфигурируемых логических блоков (Configurable Logic Block, CLB) ПЛИС семейств

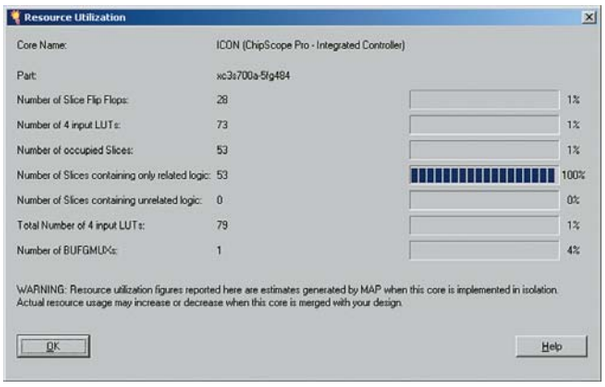

Рис. 5. Вид информационной панели, содержащей сведения о количестве различных ресурсов кристалла, используемых для автономной реализации контроллера chipscope_icon_v1_02_a

FPGA (полную версию листинга смотрите на сайте http://www.kit-e.ru/assets/listing.rar):

```
begin
  XST_GND : GND
    port map ( G => CONTROL0(2) );
  XST_VCC : VCC
    port map ( P => N1 );
  U0_U_ICON_U_STAT_U_STAT_CNT_G_5_U_FDRE : FDRE
    generic map( INIT => '0' )
    port map ( C => CONTROL0(0), CE => N1,
      D => U0_U_ICON_U_STAT_U_STAT_CNT_D(5),
      R => U0_U_ICON_U_STAT_iSTATCMD_CE_n,
      Q => U0_U_ICON_U_STAT_iSTAT_CNT(5) );
.....
.....
.....
.....
  U0_U_ICON_U_STAT_U_TDO : FDE
    generic map( INIT => '0' )
    port map ( C => CONTROL0(0), CE => N1,
      D => U0_U_ICON_U_STAT_iTDO_next,
      Q => U0_U_ICON_iTDO_VEC(15) );
end STRUCTURE;
```

Декларация сформированного контроллера *chipscope_icon_v1_02_a* в составе VHDL-описания разрабатываемого устройства осуществляется с помощью следующей конструкции.

```
component chipscope_icon_v1_02_a
  PORT (
    CONTROL0 : INOUT STD_LOGIC_VECTOR(35 DOWNTO 0);
    TDI_IN : IN STD_LOGIC;
    RESET_IN : IN STD_LOGIC;
    SHIFT_IN : IN STD_LOGIC;
    UPDATE_IN : IN STD_LOGIC;
    CAPTURE_IN : IN STD_LOGIC;
    SEL_IN : IN STD_LOGIC;
    DRCK_IN : IN STD_LOGIC;
    TDO_OUT : OUT STD_LOGIC
);
end component;
```

Для создания экземпляра сгенерированного контроллера *chipscope_icon_v1_02_a* нужно включить в состав описания отлаживаемого устройства следующий оператор:

```
<идентификатор_экземпляра_контроллера>: chipscope_icon_v1_02_a
  port map (
    CONTROL0 => CONTROL0,
    TDI_IN => TDI_IN,
    RESET_IN => RESET_IN,
    SHIFT_IN => SHIFT_IN,
    UPDATE_IN => UPDATE_IN,
    CAPTURE_IN => CAPTURE_IN,
    SEL_IN => SEL_IN,
    DRCK_IN => DRCK_IN,
    TDO_OUT => TDO_OUT
    );
```

Подробные сведения о количестве триггеров (Flip Flop), таблиц преобразования (LUT) и секций (Slices), используемых для автономной реализации контроллера *chipscope_icon_v1_02_a*, представлены в информационной панели, вид которой показан на рис. 5.

Продолжение следует

Литература

1. Зотов В. Средства внутрикристальной 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.