Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 18

Валерий ЗОТОВ
walerry@km.ru

При открытии базы данных реализованного проекта разработчику предоставляется не только доступ к дополнительным инструментам анализа результатов размещения и трассировки разрабатываемого устройства в кристалле программируемой логики или расширяемой процессорной платформы, но и возможность их оперативной коррекции. Редактирование результатов выполнения проекта в ПЛИС или программируемой системе на кристалле может потребоваться, в частности, для устранения проблем трассировки критических цепей или для достижения максимальной производительности создаваемого устройства.

Дополнительные возможности анализа результатов размещения и трассировки разрабатываемого устройства в кристалле

Открытие базы данных реализованного в кристалле проекта осуществляется командой **Open Implemented Design**, которая одновременно представлена в разделе Implementation панели управления основными процессами Flow Navigator основного окна интегрированной среды разработки Vivado Integrated Design Environment (IDE) и во всплывающем меню **Flow**. При выборе указанной команды во всплывающем меню Flow становится доступным меню следующего уровня, где перечислены идентификаторы всех завершенных вариантов процесса размещения и трассировки проекта в кристалле программируемой логики или расширяемой вычислительной платформы. После выбора интересующего варианта на экране появляется информационная панель с заголовком **Load Implemented Design**, содержащая краткие сведения о процессе загрузки, инициализации и подготовке к использованию базы данных реализованного в кристалле проекта (рис. 1).

По окончании указанного процесса основное окно управляющей оболочки САПР серии Xilinx Vivado HLx Design Suite принимает вид, представленный на рис. 2.

В области расположения окон данных проекта появляется вкладка **Device Constraints**, включающая одноименное встроенное окно, которое предоставляет возможность установки и редактирования физических ограничений, определяющих параметры конфигурирования выводов кристалла программируемой логики или расширяемой процессорной платформы, применяемого для реализации разрабатываемого устройства. В рабочей области основного окна интегрированной среды разработки Vivado IDE открывается вкладка **Package**, которая содержит графическое интерактивное представление топологии выводов корпуса ПЛИС или программируемой системы на кристалле с корпусом, выбранного для реализации проектируемого устройства. Консольная область основного окна управляющей оболочки САПР серии Xilinx Vivado HLx Design Suite представляет вкладку с заголовком *Timing-Timing Summary -<идентификатор_рассматриваемого_варианта_процесса_реализации_проекта_в_кристалле>*, в которой находится сводный отчет о временных параметрах реализованного проекта. В разделе Implementation панели управления Flow Navigator основного окна управляющей оболочки САПР серии Xilinx Vivado HLx Design Suite открывается подраздел Implemented Design, обеспечивающий быстрый доступ к основным инструментам углубленного анализа результатов этапа размещения и трассировки проектируемого устройства в кристалле программируемой логики или расширяемой процессорной платформы.

После открытия базы данных реализованного в кристалле проекта разработчик получает следующие возможности:

- проведение наиболее точного временного анализа проекта с формированием отчетов о различных временных параметрах сигналов разрабатываемого устройства;
- осуществление контроля выполнения рекомендаций, предписываемых методологией проектирования высокопроизводительных цифровых устройств на базе ПЛИС и программируемых систем на кристалле последних поколений и установленных проектных норм;
- коррекция временных и топологических ограничений проекта с помощью соответствующих интерактивных средств интегрированной среды разработки Vivado IDE;
- выполнение детализированного анализа энергопотребления разрабатываемого устройства с генерацией соответствующего интерактивного отчета;
- формирование рекомендаций по оптимизации потребляемой мощности проектируемого устройства;
- визуальная оценка результатов размещения элементов проектируемого устройства в кристалле программируемой логики или расширяемой процессорной платформы;
- визуальный контроль трассировки цепей проектируемого устройства в кристалле;
- анализ трассировки цепей с большим количеством разветвлений;
- генерация интерактивного отчета о конфигурировании выводов кристалла программируемой логики или расширяемой процессорной платформы, реализующего функции проектируемого устройства.

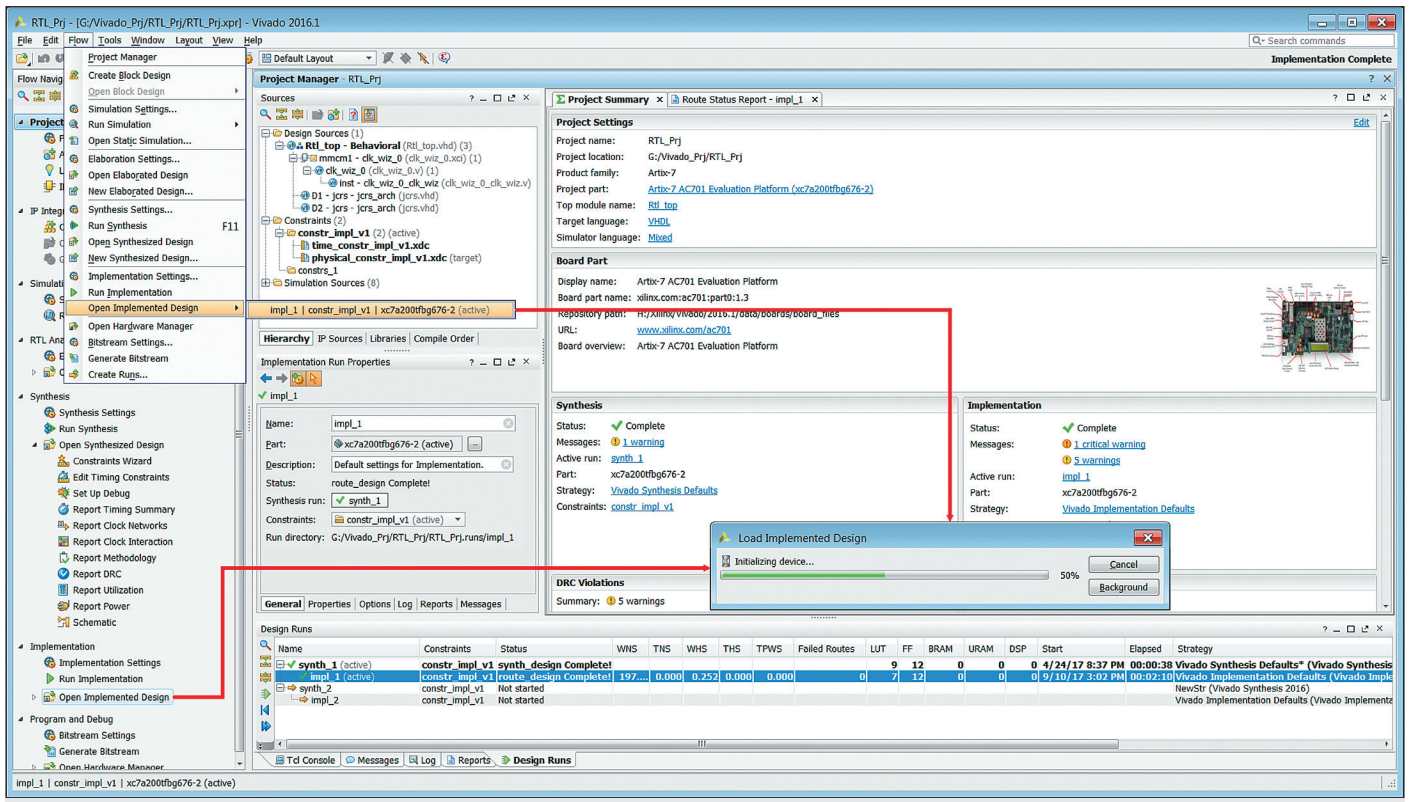

Рис. 1. Открытие базы данных реализованного в кристалле проекта

Уточненный анализ временных характеристик и энергопотребления проектируемого устройства, проверка выполнения рекомендаций методологии UltraFast Design Methodology [24, 25] и проектных норм выполняются таким же образом, как и предварительный анализ, производимый при открытии базы данных синтезированного проекта [26–30]. Поэтому далее основное внимание уделено визуальному контролю результатов размещения и трассировки проекта в кристалле программируемой логики или расширяемой процессорной платформы.

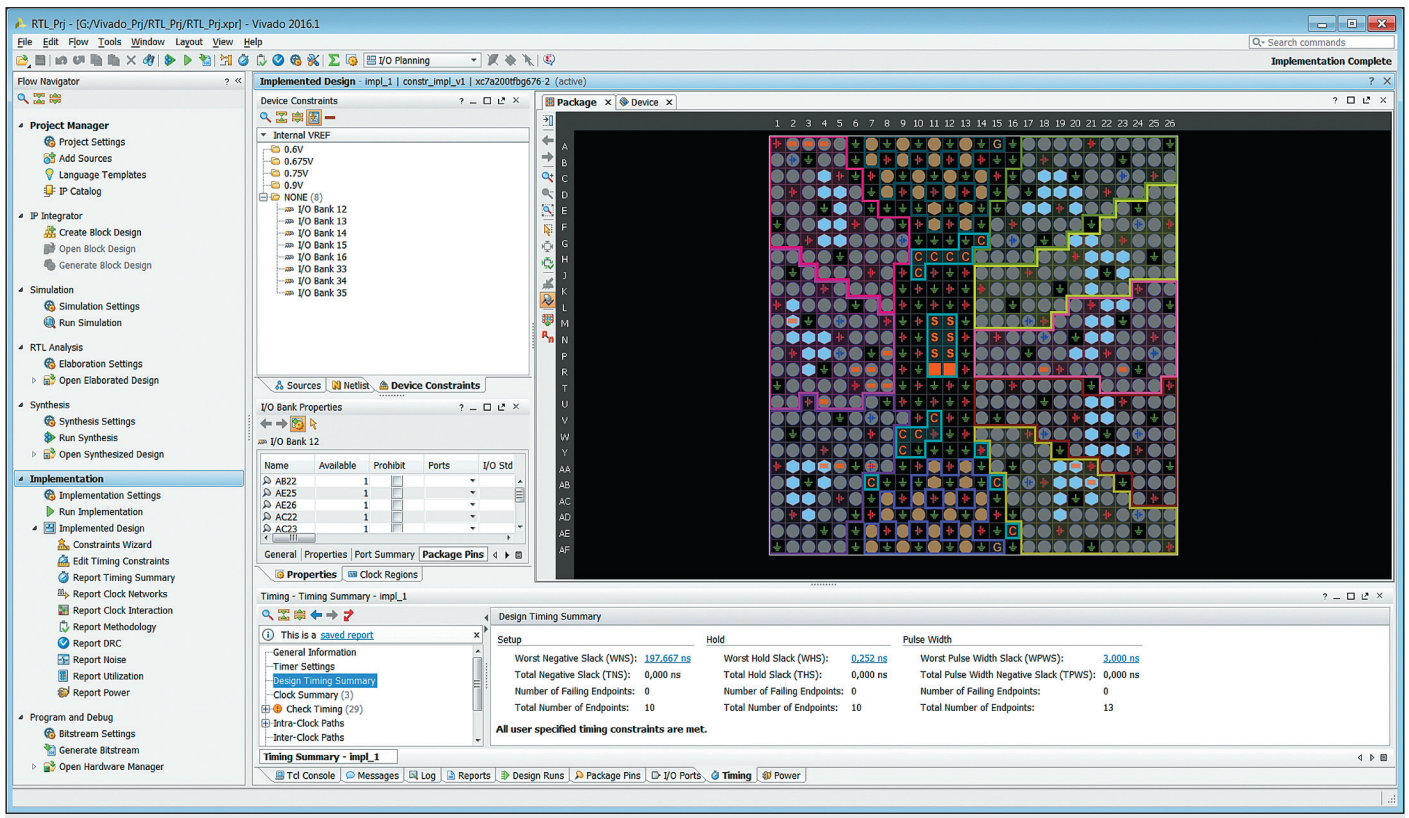

Рис. 2. Вид основного окна интегрированной среды разработки Vivado IDE после открытия базы данных реализованного в кристалле проекта

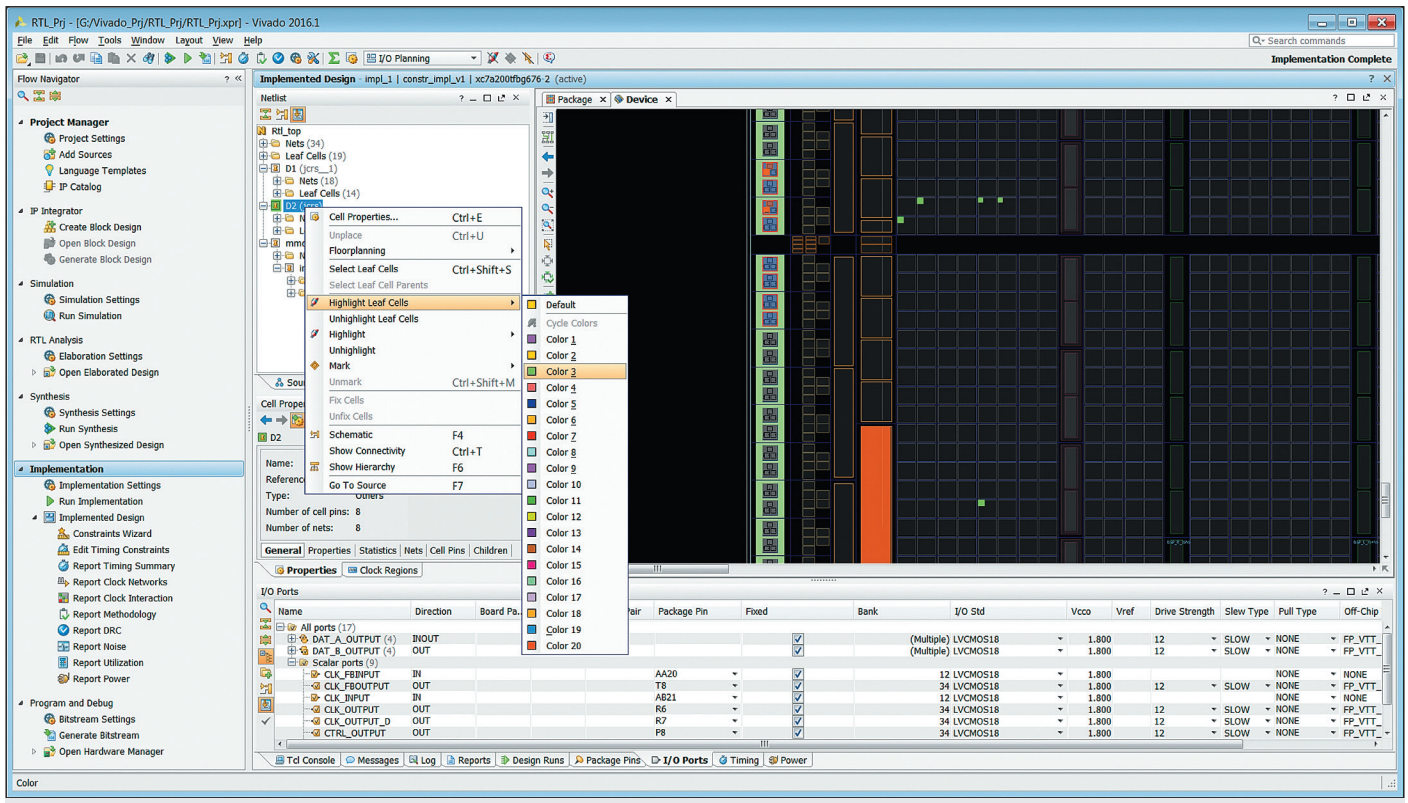

Рис. 3. Отображение размещения элементов проектируемого устройства в кристалле программируемой логики или расширяемой процессорной платформы

Визуальная оценка размещения элементов проектируемого устройства в кристалле программируемой логики или расширяемой процессорной платформы

Для визуальной оценки размещения элементов проектируемого устройства в кристалле программируемой логики или расширяемой процессорной платформы нужно прежде всего в рабочей области основного окна управляющей оболочки САПР серии Xilinx Vivado HLx Design Suite открыть вкладку **Device**. Затем в области расположения окон данных проекта сделать активной вкладку **Netlist**, на которой отображается иерархическая структура реализуемого списка соединений разрабатываемого устройства. В этом списке следует указать курсором и выделить щелчком левой кнопки мыши строку, содержащую идентификатор интересующего компонента иерархической структуры. После этого нужно щелчком правой кнопки мыши вызвать контекстно зависимое всплывающее меню, как показано на рис. 3. В открывшемся меню надо воспользоваться командой **Highlight Leaf Cells**, при выборе которой появляется всплывающее меню следующего уровня, чей вид представлен на рис. 3. В этом меню надо указать цвет для выделения условных обозначений физических ресурсов ПЛИС или программируемой системы на кристалле, задействованных для реализации выбранного иерархического элемента списка соединений проектируемого устройства.

Если на вкладке **Netlist** отмечены идентификаторы нескольких элементов иерархической структуры представленного списка соединений, то во всплывающем меню выбора цвета выделения нужно указать вариант Cycle Colors. После закрытия меню условные графические образы (УГО) физических ресурсов кристалла программируемой логики или расширяемой процессорной платформы, используемых для реализации обособленного элемента списка соединений проектируемого устройства, будут подсвечены на вкладке **Device** выбранным цветом. Для их детального отображения рекомендуется увеличить масштаб изображения на указанной вкладке. Таким образом, назначив различные цвета для выделения УГО физических ресурсов, применяемых для реализации соответствующих компонентов иерархической структуры списка соединений разрабатываемого устройства, можно получить наглядную картину их размещения в кристалле.

При анализе размещения элементов проектируемого устройства в ПЛИС или программируемой системе на кристалле целесообразно отобразить и расположение используемых ресурсов кристалла, сопряженных с выводами выбранного компонента реализуемого списка соединений разрабатываемого устройства. В частности, можно проанализировать топологию задействованных ресурсов кристалла, определяющих состояние сигналов на входах интересующего компонента. Для этой цели в контекстно зависимом всплывающем меню предусмотрена команда **Show Connectivity**, при выборе которой на изображении топологической структуры кристалла программируемой логики или расширяемой процессорной платформы, отображаемом на вкладке **Device**, выделяются УГО соответствующих физических ресурсов. Взаимодействие этих ресурсов с входами и выходами анализируемого элемента списка соединений проектируемого устройства изображено стрелками (рис. 4).

Вкладка **Device** рабочей области основного окна интегрированной среды разработки Vivado IDE предоставляет и возможность непосредственного детализированного анализа взаимодействия физических ресурсов ПЛИС или программируемой системы на кристалле, применяемых для реализации проектируемого устройства. Для этого вначале необходимо выделить УГО интересующего ресурса кристалла, задействованного в составе проекта, указав его курсором и щелкнув левой кнопкой мыши. Затем, расположив курсор на вкладке Device, щелчком правой кнопки мыши нужно вызвать контекстно зависимое всплывающее меню и выбрать в нем строку Show Connectivity (рис. 5). После выполнения данной команды на вкладке отображается взаимодействие выделенного элемента с другими ресурсами кристалла, используемыми для реализации проектируемого устройства.

В контекстно зависимом всплывающем меню, доступном после выделения УГО ресурса кристалла, задействованного для реализации

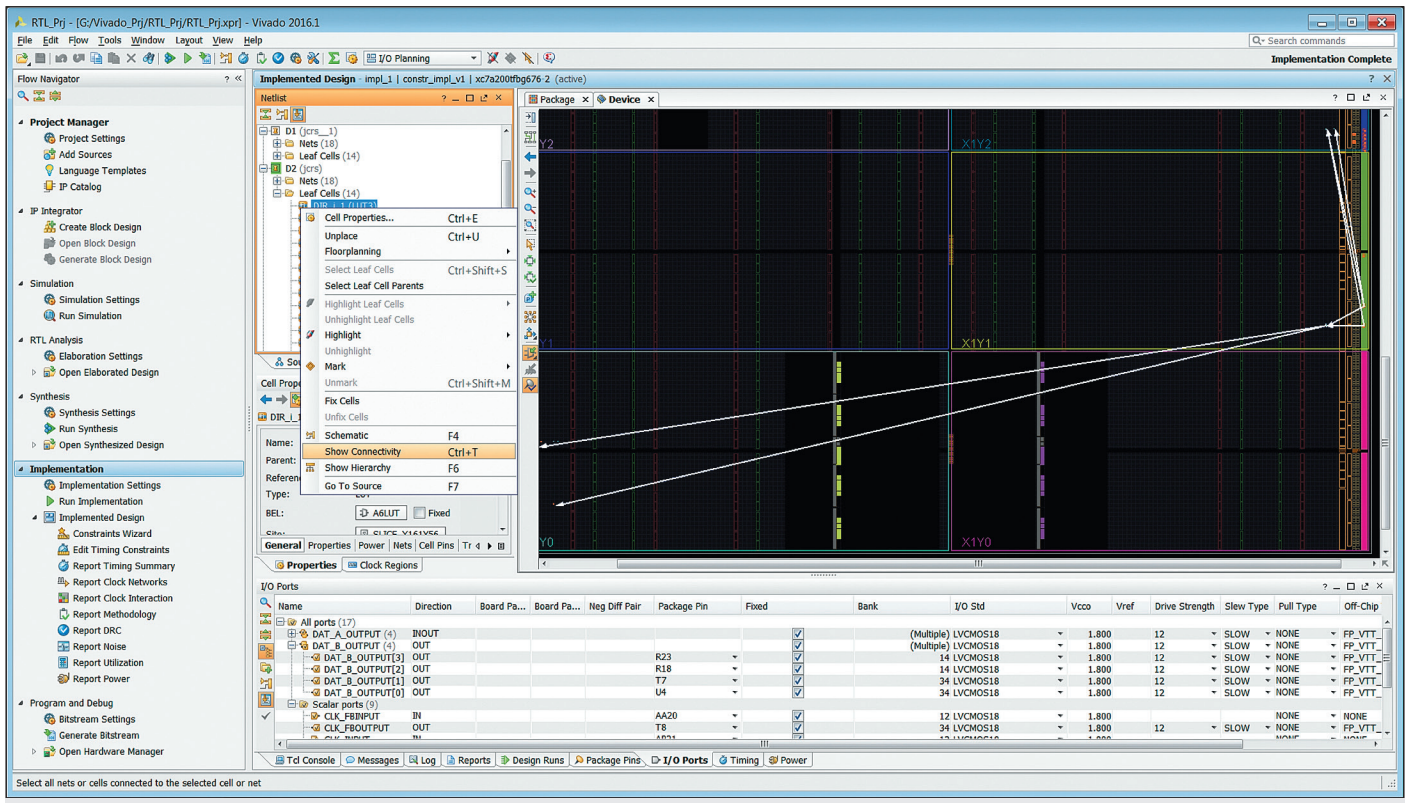

Рис. 4. Отображение расположения используемых ресурсов кристалла, которые сопряжены с выводами выбранного компонента списка соединений

проекта, представлена также команда **Go To Source** (рис. 5), позволяющая проследить соответствие исходного кода HDL-описания разрабатываемого устройства и используемых физических ресурсов ПЛИС или программируемой системы на кристалле. Эта команда отображает код исходного HDL-описания разрабатываемого устройства, соответствующего выделенному УГО. Аналогичная команда содержится в контекстно зависимом всплывающем меню, открываемом после выделения строки с идентификатором интересующего элемента на вкладке **Netlist**, чей вид приведен на рис. 6. После выбора команды **Go To Source** в рабочей области основного окна управляю-

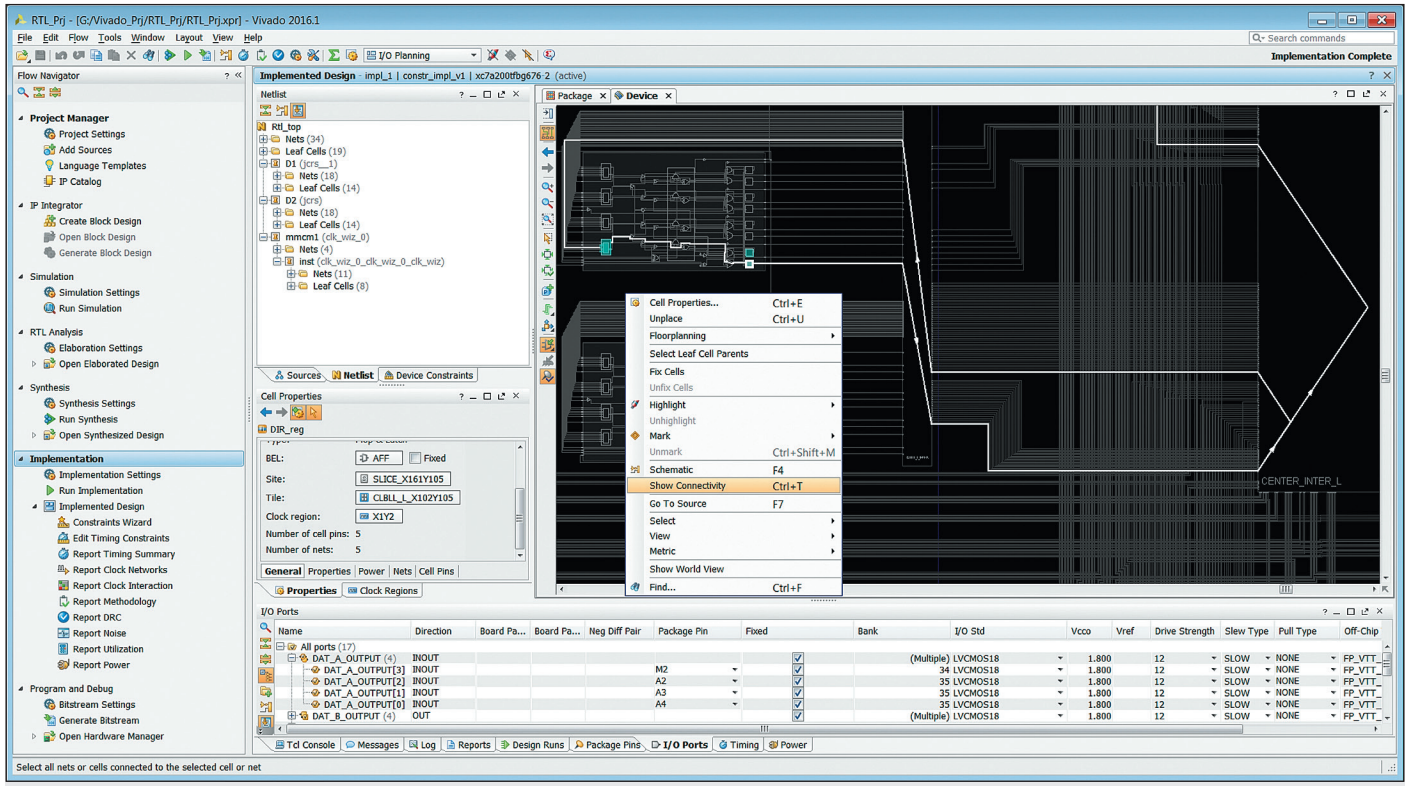

Рис. 5. Отображение топологии используемых ресурсов кристалла, которые сопряжены с выбранным элементом архитектуры кристалла, задействованным в проекте

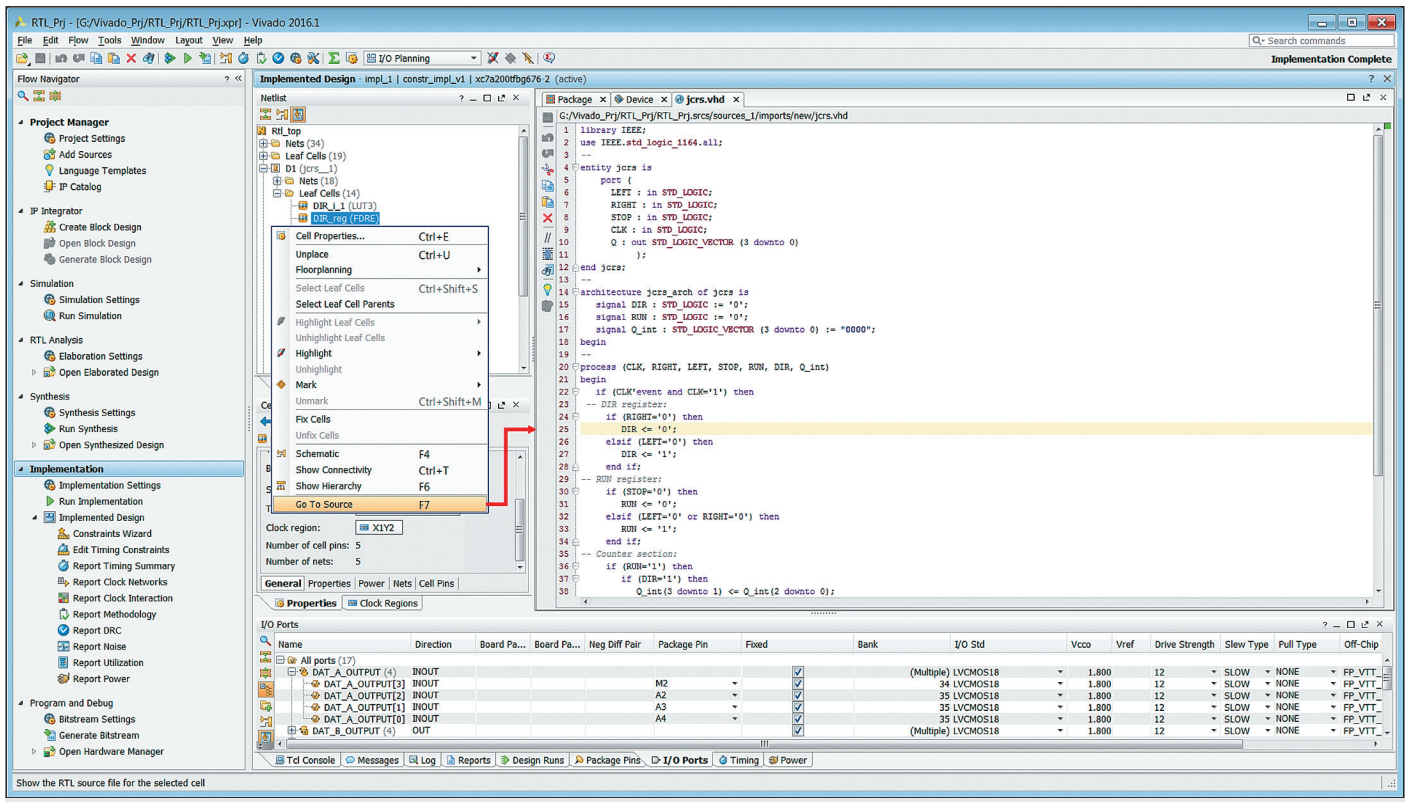

Рис. 6. Отображение кода исходного HDL-описания разрабатываемого устройства, которое соответствует выделенному элементу архитектуры кристалла, задействованному в проекте

щей оболочки САПР серии Xilinx Vivado HLx Design Suite добавляется новая вкладка окна встроенного HDL-редактора, где представлен текст надлежащего модуля исходного описания проекта. В этом окне выделена строка HDL-кода, соответствующая выбранному ресурсу кристалла программируемой логики или расширяемой процессорной платформы (рис. 6).

При разработке сложных проектов рекомендуется провести детализированный анализ плотности распределения логических ресурсов ПЛИС или программируемой системы на кристалле, применяемых для их реализации. В результате такого анализа можно выявить перегруженные области кристалла программируемой логики или расширяемой вычислительной платформы, способные привести к проблемам в процессе трассировки цепей проекта. Для этой цели в САПР серии Xilinx Vivado HLx Design Suite предусмотрена система показателей, в наглядной форме отображающая результаты детализированного анализа плотности распределения используемых логических ресурсов ПЛИС или программируемой системы на кристалле.

Детализированный анализ плотности распределения логических ресурсов ПЛИС или программируемой системы на кристалле

Для осуществления детализированного анализа плотности распределения в кристалле программируемой логики или расширяемой вычислительной платформы ресурсов, применяемых в составе проектируемого устройства, целесообразно открыть встроенное окно **Metrics**. Оно предоставляет доступ к различным показателям, характеризующим плотность распределения логических ресурсов, задействованных для реализации разрабатываемого проекта. Вкладка окна **Metrics** открывается в области расположения окон данных проекта интегрированной среды разработки Vivado IDE при выборе команды **Metrics** во всплывающем меню **Window** (рис. 7).

В открывшемся встроенном окне приведен список названий показателей, характеризующих плотность распределения используемых логических ресурсов кристалла программируемой логики или расширяемой вычислительной платформы, которые разделены на четыре группы — Primitive Metrics, CLB Metrics, Clock Region Metrics и Pblock Metrics. Группа Primitive Metrics включает единственный показатель Min Slack per placed BEL, позволяющий анализировать используемые ресурсы ПЛИС или программируемой системы на кристалле с минимальным запасом по времени.

В состав группы CLB Metrics входят характеристики применения ресурсов конфигурируемых логических блоков для реализации разрабатываемого проекта. Показатели LUT Utilization per CLB и FF Utilization per CLB предусматривают анализ соответственно количества таблиц преобразования Look-Up Table (LUT) и триггеров каждого конфигурируемого логического блока, задействованных в проектируемом устройстве, выраженного в процентном отношении к общему объему этих ресурсов CLB. С помощью показателей Vertical Routing Congestion per CLB и Horizontal Routing Congestion per CLB можно оценить перегруженность соответственно вертикальных и горизонтальных трассировочных ресурсов для каждого конфигурируемого логического блока ПЛИС или программируемой системы на кристалле.

Группа Clock Region Metrics объединяет характеристики использования различных логических ресурсов для реализации разрабатываемого проекта в каждом регионе тактирования. Показатели LUT Utilization per Clock Region и FF Utilization per Clock Region предоставляют возможность анализа количества таблиц преобразования LUT и триггеров, задействованных в составе проектируемого устройства в различных регионах тактирования ПЛИС или программируемой системы на кристалле.

В группе Pblock Metrics представлены характеристики, информирующие об объеме различных логических ресурсов, применяемых в составе P-блоков. Показатели LUT Utilization per Pblock и FF Utilization per Pblock помогают оценить соответственно количество таблиц преобразования LUT и триггеров, включенных в P-блоки, сформированные в разрабатываемом проекте.

Прежде чем приступить к выполнению анализа плотности распределения в кристалле программируемой логики или расширяемой

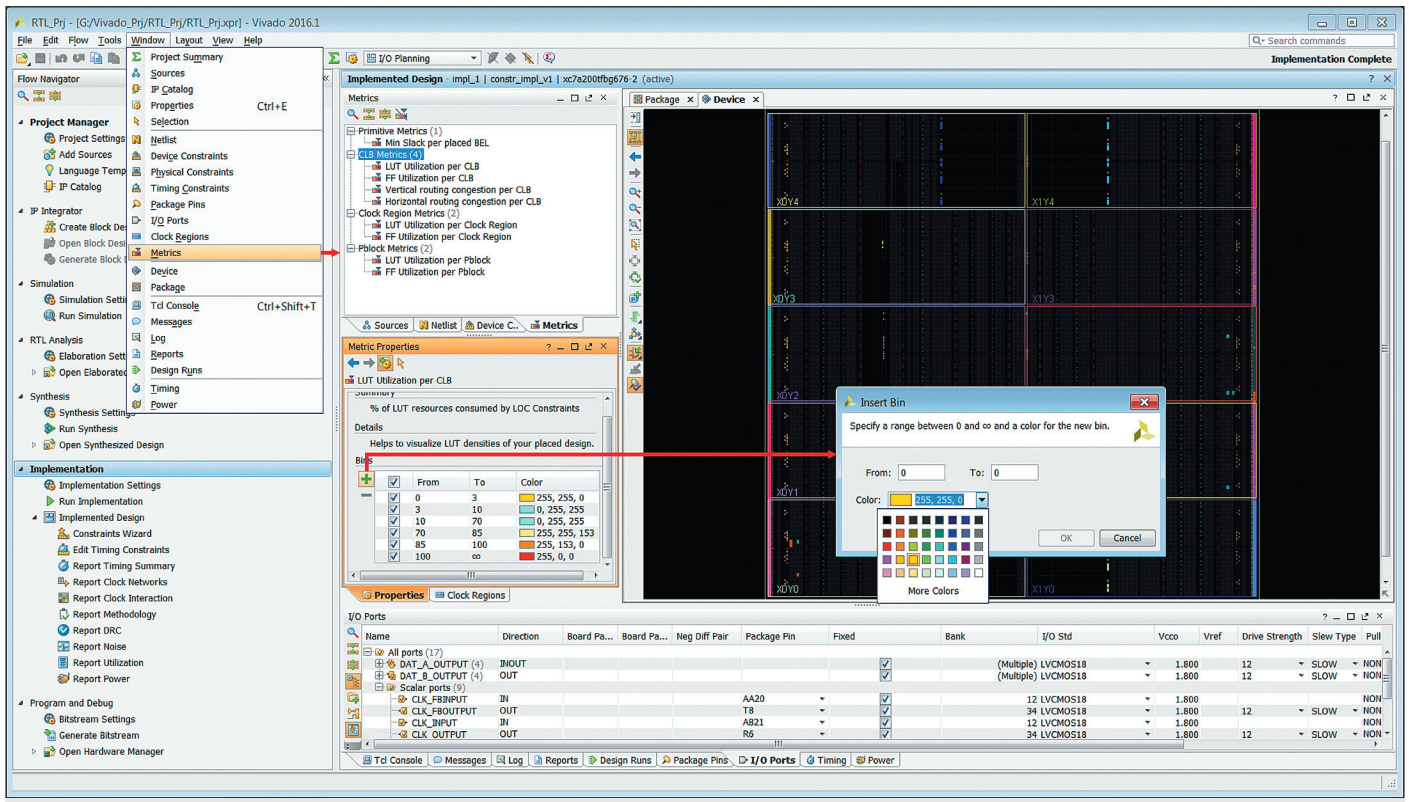

Рис. 7. Открытие окна анализа показателей, характеризующих плотность распределения используемых логических ресурсов кристалла

вычислительной платформы ресурсов, задействованных для реализации проектируемого устройства, рекомендуется ознакомиться и при необходимости изменить параметры отображения вычисляемых показателей. Для этого следует во встроенном окне **Metrics** выделить строку с названием интересующего показателя, расположив на ней курсор и щелкнув левой кнопкой мыши. После выполнения указанных действий во встроенном окне **Metric Properties**, которое также открывается при выборе команды **Metrics** всплывающего меню **Window**, появляется таблица параметров отображения выделенного показателя, чей вид изображен на рис. 7. В каждой строке таблицы приведены значения параметров соответствующего интервала величин выбранного показателя. В колонках From и To представлены начальные и конечные границы интервалов, на которые разбит весь диапазон значений выбранного показателя. Колонка Color содержит информацию о цветах выделения УГО логических ресурсов на вкладке **Device**, согласно каждому интервалу значений выбранного показателя. В первой колонке таблицы представлены индикаторы состояния, предназначенные для управления выделением УГО логических ресурсов, которые относятся к каждому интервалу. При включенном состоянии индикатора соответствующие УГО будут выделены на вкладке Device цветом, указанным в колонке Color. Таким образом, предоставляется возможность отображения только используемых логических ресурсов, отвечающих нужным значениям выбранного показателя.

Для изменения интервалов, на которые разбит диапазон значений интересующего показателя, следует воспользоваться кнопками, расположенными на панели инструментов встроенного окна **Metric Properties**. Для добавления нового интервала значений служит кнопка ➕. При ее нажатии на экране появляется диалоговая панель с названием Insert Bin (рис. 7). Открывшаяся диалоговая панель содержит поля редактирования значений параметров From и To, в которых с помощью клавиатуры следует указать начальную и конечную границу добавляемого интервала. Затем, используя поле выбора значения параметра Color, необходимо определить цвет, которым будут выделяться УГО логических ресурсов на вкладке **Device**, соответствующих формируемому интервалу. При нажатии кнопки управления выпадающим списком значений этого поля выбора открывается панель, где представлена палитра доступных цветов. Нужный цвет выделения достаточно указать курсором в палитре и щелкнуть левой кнопкой мыши. Определив требуемые значения параметров From, To и Color, необходимо подтвердить их нажатием клавиши **OK**, расположенной в нижней части диалоговой панели **Insert Bin**. После закрытия диалоговой панели в таблице, отображаемой во встроенном окне **Metric Properties**, появляется новая строка, где приведены установленные параметры сформированного интервала значений. При этом выполняется автоматическая корректировка границ соседних интервалов. Удаление неактуального интервала значений выбранного показателя выполняется нажатием кнопки ➖ после выделения соответствующей строки таблицы.

Для отображения выбранного показателя плотности распределения используемых ресурсов в кристалле программируемой логики или расширяемой процессорной платформы нужно после выделения соответствующей строки во встроенном окне **Metrics** щелчком правой кнопки мыши вызвать контекстно зависимое всплывающее меню и выбрать в нем команду **Show** (рис. 8).

После выполнения указанной команды УГО конфигурируемых логических блоков, представленные на вкладке **Device** на изображении топологической структуры применяемого кристалла, выделяются цветом, соответствующим интервалу значений выбранного показателя, характеризующего плотность распределения логических ресурсов ПЛИС или программируемой системы на кристалле, задействованных в проекте. Таким образом, разработчику предоставляется возможность наглядной визуальной оценки областей кристалла программируемой логики или расширяемой процессорной платформы с повышенной плотностью ресурсов, применяемых для реализации проектируемого устройства. Кроме того, в консольной области основного окна управляющей оболочки САПР серии Xilinx Vivado HLx Design Suite открывается вкладка с названием **Metric Results**, предоставляющая более точные сведения о числовых значениях анализируемого показателя. Приведенная здесь информация отображается

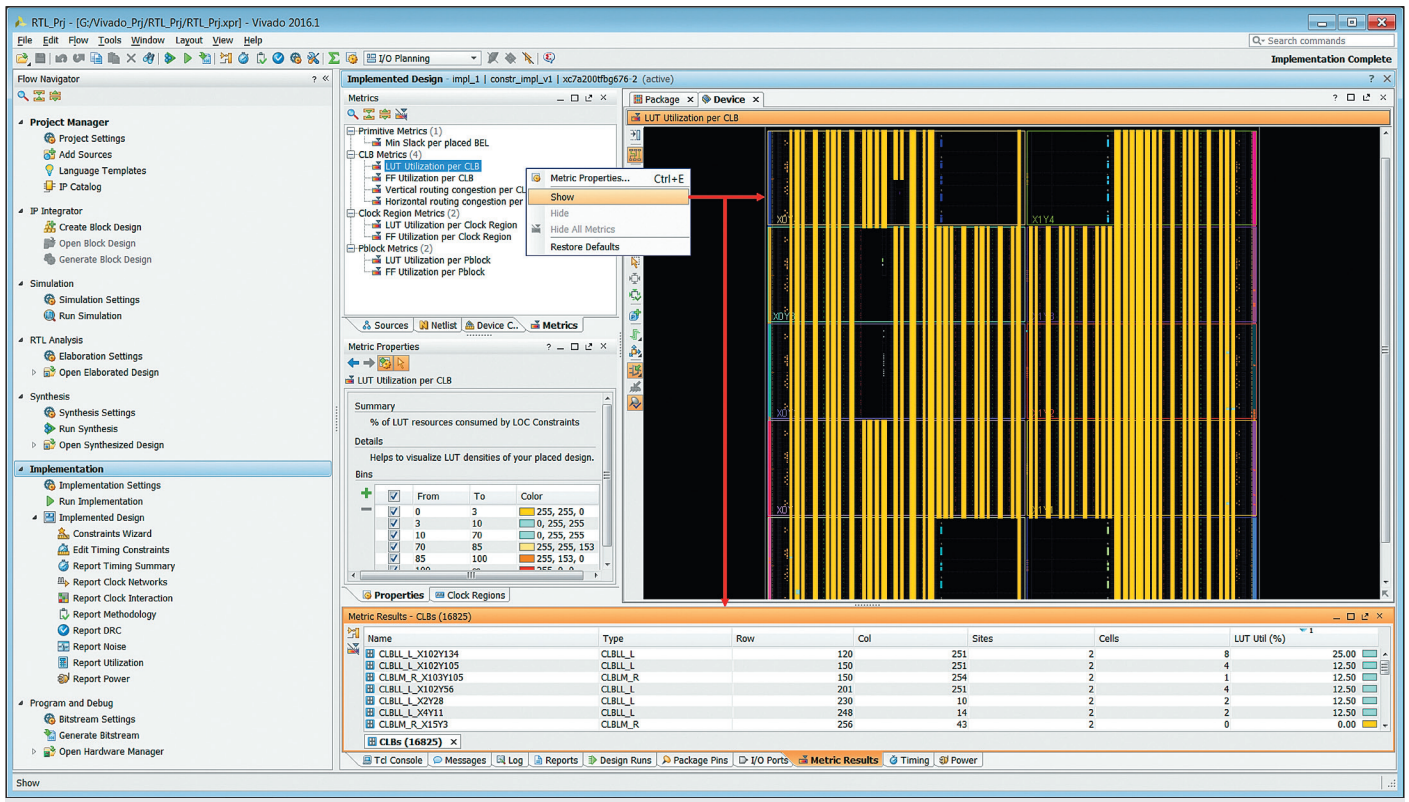

Рис. 8. Анализ выбранного показателя плотности распределения используемых ресурсов в кристалле программируемой логики или расширяемой процессорной платформы

в форме таблицы, чья структура зависит от выбранного показателя. В частности, при анализе характеристик LUT Utilization per CLB и FF Utilization per CLB в соответствующих колонках таблицы дано условное обозначение каждого конфигурируемого логического блока (Name), его тип (Type) и координаты расположения в кристалле (Row, Col, Sites), количество ячеек (Cells) и процент используемых различных ресурсов Lut Util (%) или FF Util (%).

Для быстрого отображения интересующих показателей, характеризующих плотность распределения логических ресурсов ПЛИС или программируемой системы на кристалле, можно воспользоваться

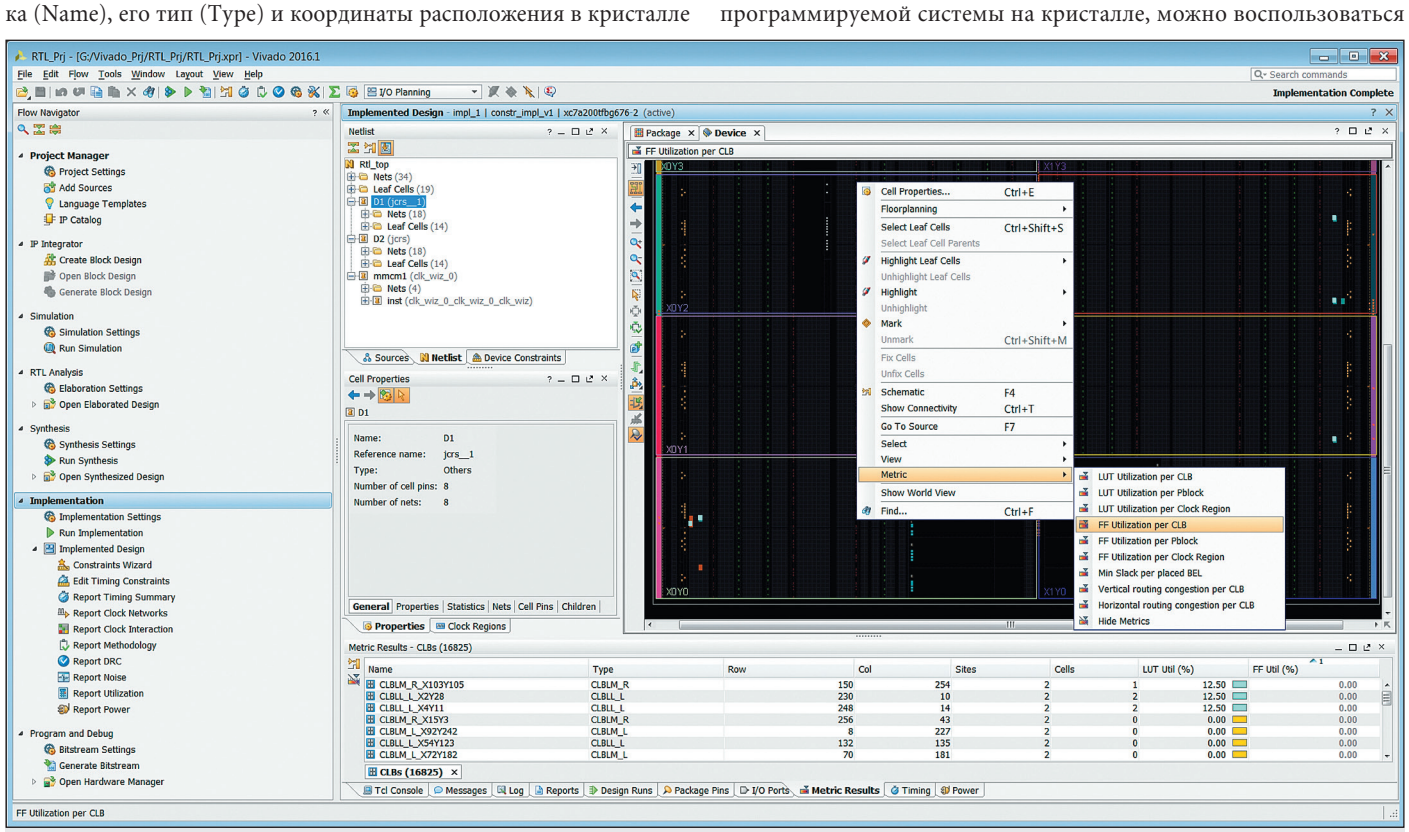

Рис. 9. Быстрое отображение выбранного показателя плотности распределения используемых ресурсов ПЛИС или программируемой системы на кристалле

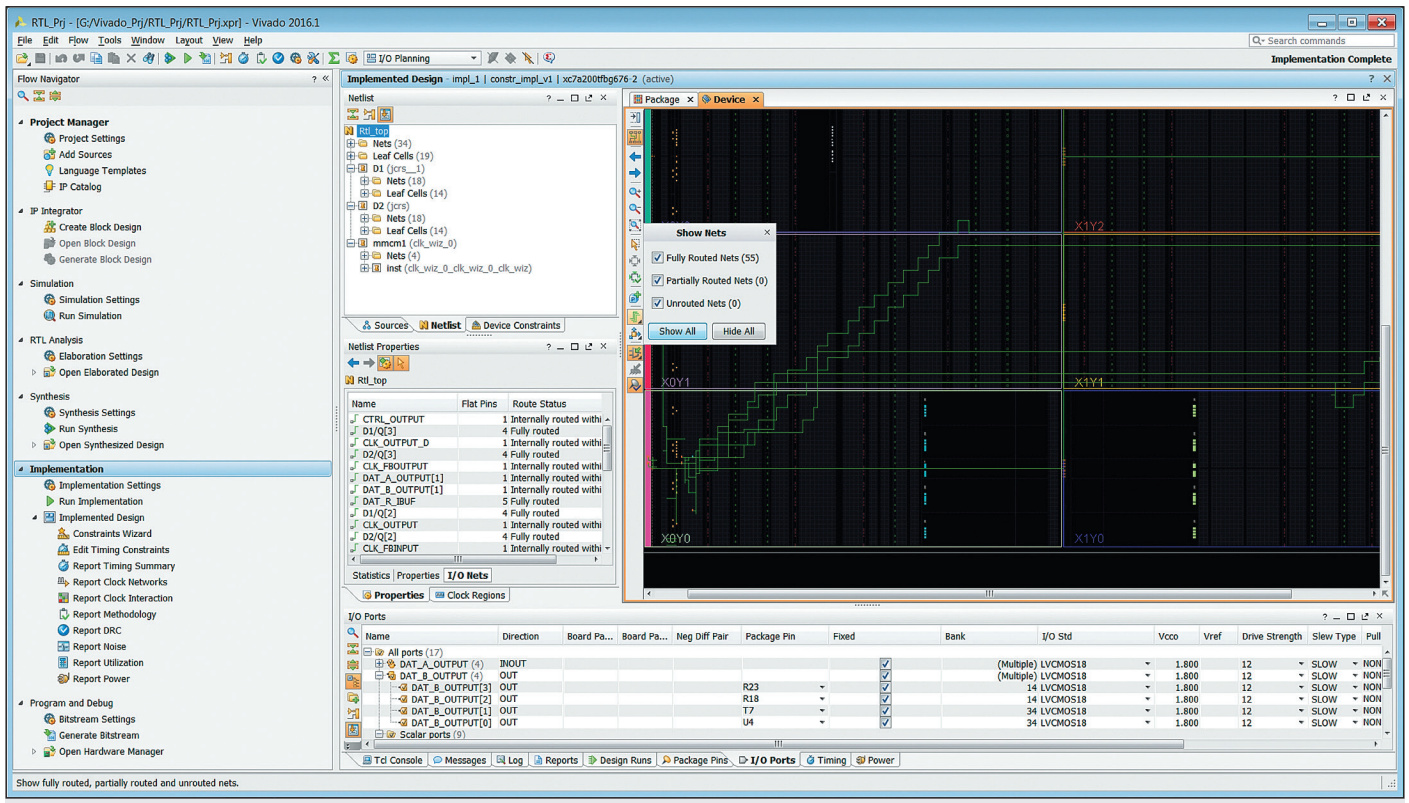

Рис. 10. Отображение трассировки цепей проектируемого устройства в кристалле программируемой логики или расширяемой процессорной платформы

контекстно зависимым всплывающим меню, вызываемым щелчком правой кнопки мыши при расположении курсора на вкладке **Device**. В меню имеется команда **Metric**, при выборе которой открывается всплывающее меню следующего уровня, чей вид приведен на рис. 9. Открывшееся меню включает список названий показателей, рассмотренных выше, а также команду **Hide Metric**. Сразу после выбора в этом меню строки, соответствующей интересующему показателю, информация о плотности распределения используемых ресурсов кристалла программируемой логики или расширяемой процессорной платформы будет представлена в графическом виде на вкладке **Device** и в числовой форме на вкладке **Metric Results**. При этом будут использоваться параметры отображения вычисляемых показателей, установленные ранее или предлагаемые по умолчанию. Отключение графического представления информации о плотности распределения логических ресурсов ПЛИС или программируемой системы на кристалле, задействованных для реализации проектируемого устройства, осуществляется командой **Hide Metric**.

Визуальный контроль результатов трассировки цепей проектируемого устройства в кристалле программируемой логики или расширяемой процессорной платформы

Вкладка **Device** рабочей области основного окна интегрированной среды разработки Vivado IDE позволяет в наглядной форме отобразить результаты трассировки цепей проектируемого устройства в кристалле программируемой логики или расширяемой вычислительной платформы. Чтобы воспользоваться этой возможностью, следует зафиксировать в нажатом состоянии кнопку на панели инструментов указанной вкладки. После этого на изображении топологической структуры кристалла появляется информация о трассировочных ресурсах ПЛИС или программируемой системы на кристалле. Затем нужно переключить в нажатое положение кнопку , также имеющуюся на панели инструментов вкладки **Device**. Откроется диалоговая панель с заголовком Show Nets (рис. 10).

Указанная диалоговая панель позволяет выбрать категории цепей, информация о которых будет представлена на изображении топологической структуры кристалла программируемой логики или расширяемой процессорной платформы. Для отображения полностью разведенных цепей необходимо установить в положение «Включено» индикатор состояния параметра **Fully Routed Nets**. Чтобы показать частично разведенные цепи, нужно перевести во включенное положение индикатор состояния параметра Partially Routed Nets. Для представления наглядной информации о неразведенных цепях проектируемого устройства следует установить в положение «Включено» индикатор состояния параметра Unrouted Nets. После нажатия клавиши **Show All**, расположенной в нижней части диалоговой панели Show Nets, информация о трассировке всех цепей выбранных категорий будет отражена на вкладке **Device**, как показано на рис. 10.

Анализ результатов трассировки цепей проектируемого устройства с большим количеством разветвлений

Для того чтобы проанализировать результаты трассировки цепей проектируемого устройства с большим количеством разветвлений, надо вначале воспользоваться командой **Report** из всплывающего меню **Tools**. На экране появится всплывающее меню следующего уровня, в котором нужно выбрать команду **Report High Fanout Nets** (рис. 11). В процессе выполнения данной команды формируется отчет о цепях проектируемого устройства, имеющих большое количество разветвлений. Отчет представлен на вкладке High Fanout Nets, которая автоматически открывается в консольной области основного окна управляющей оболочки САПР серии Xilinx Vivado HLx Design Suite после завершения команды Report High Fanout Nets.

Сгенерированный отчет имеет два раздела с названиями General Information и Summary. В разделе General Information приведена общая информация о разрабатываемом проекте, типе ПЛИС или программируемой системы на кристалле, используемой для

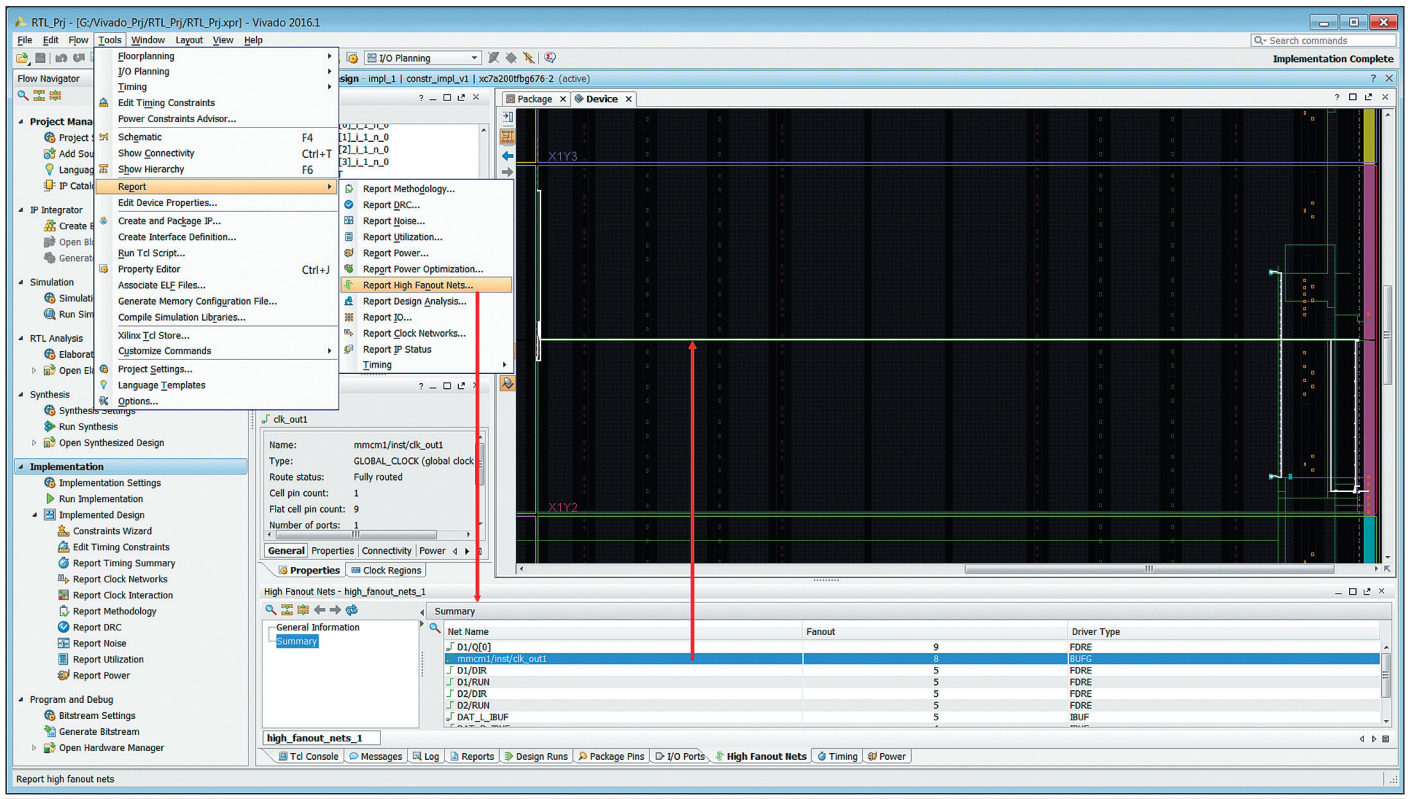

Рис. 11. Анализ результатов трассировки цепей проектируемого устройства с большим количеством разветвлений

его реализации, версии САПР серии Xilinx Vivado HLx Design Suite, в среде которой сформирован отчет, дате и времени создания отчета. Здесь же представлен полный формат команды языка TCL, предназначенной для генерации рассматриваемого отчета, который включает все установленные параметры.

Раздел Summary содержит подробные данные о цепях проектируемого устройства с большим количеством разветвлений. Сведения отображаются в форме таблицы, разбитой на три колонки с названиями Net Name, Fanout и Driver Type. В столбце Net Name перечислены идентификаторы цепей проектируемого устройства, которые имеют большое количество разветвлений. Ячейки колонки Fanout предоставляют информацию о числе разветвлений соответствующих цепей. В ячейках столбца Driver Type указан тип элемента, формирующего сигнал в соответствующих цепях.

Для изучения результатов трассировки каждой цепи с большим количеством разветвлений достаточно выделить нужную строку рассмотренной таблицы, расположив в ней курсор и щелкнув левой кнопкой мыши. Интересующая цепь будет выделена на изображении топологической структуры кристалла программируемой логики или расширяемой процессорной платформы (рис. 11).

Оценка дополнительных возможностей минимизации энергопотребления разрабатываемого устройства

Кроме уточненного анализа энергопотребления разрабатываемого устройства с генерацией соответствующего интерактивного отчета после открытия базы данных реализованного в кристалле проекта САПР серии Xilinx Vivado HLx Design Suite предоставляет дополнительные возможности минимизации потребляемой мощности. Для их оценки следует изучить рекомендации по оптимизации потребляемой мощности проектируемого устройства, формируемые интегрированной средой разработки Vivado IDE, а также проанализировать интенсивность переключений управляющих сигналов, влияющих на уровень энергопотребления.

Для создания отчета, содержащего рекомендации по оптимизации потребляемой мощности разрабатываемого устройства, нужно во всплывающем меню **Tools** выделить строку Report, после чего на экране появляется всплывающее меню следующего уровня, в котором надо выбрать команду **Report Power Optimization**. В процессе выполнения команды открывается одноименная диалоговая панель, чей вид изображен на рис. 12.

В открывшейся диалоговой панели целесообразно в первую очередь в поле редактирования значения параметра Results name уточнить название создаваемого отчета. По умолчанию в этом поле отображается идентификатор *power_opt_N*, где *N* — порядковый номер генерируемого отчета, предоставляющего рекомендации по оптимизации энергопотребления проектируемого устройства. При создании нескольких вариантов отчета следует в поле редактирования значения параметра Results name с помощью клавиатуры задать мнемоническое название каждого генерируемого отчета. При необходимости сохранить сформированный отчет в виде файла на диске нужно установить в положение «Включено» индикатор состояния параметра Export to file, а затем указать формат, место расположения и идентификатор создаваемого файла. Определить формат файла отчета можно с помощью группы кнопок с зависимой фиксацией Output file format. Для сохранения отчета в виде текстового файла следует зафиксировать в нажатом состоянии кнопку **TXT**. Чтобы сохранить сгенерированный отчет в формате XML, нужно переключить в нажатое положение одноименную кнопку. Для выбора диска и раздела для записи файла отчета, а также определения его названия целесообразно воспользоваться стандартной диалоговой панелью сохранения файла, которая открывается при нажатии кнопки с пиктограммой в виде многоточия (рис. 12).

Для того чтобы сформированный отчет был представлен на новой странице соответствующей вкладки, надо установить индикатор состояния параметра Open in a new tab во включенное положение. После определения значений параметров, представленных в диалоговой панели Report Power Optimization, необходимо подтвердить их нажатием клавиши **OK**, находящейся в нижней части

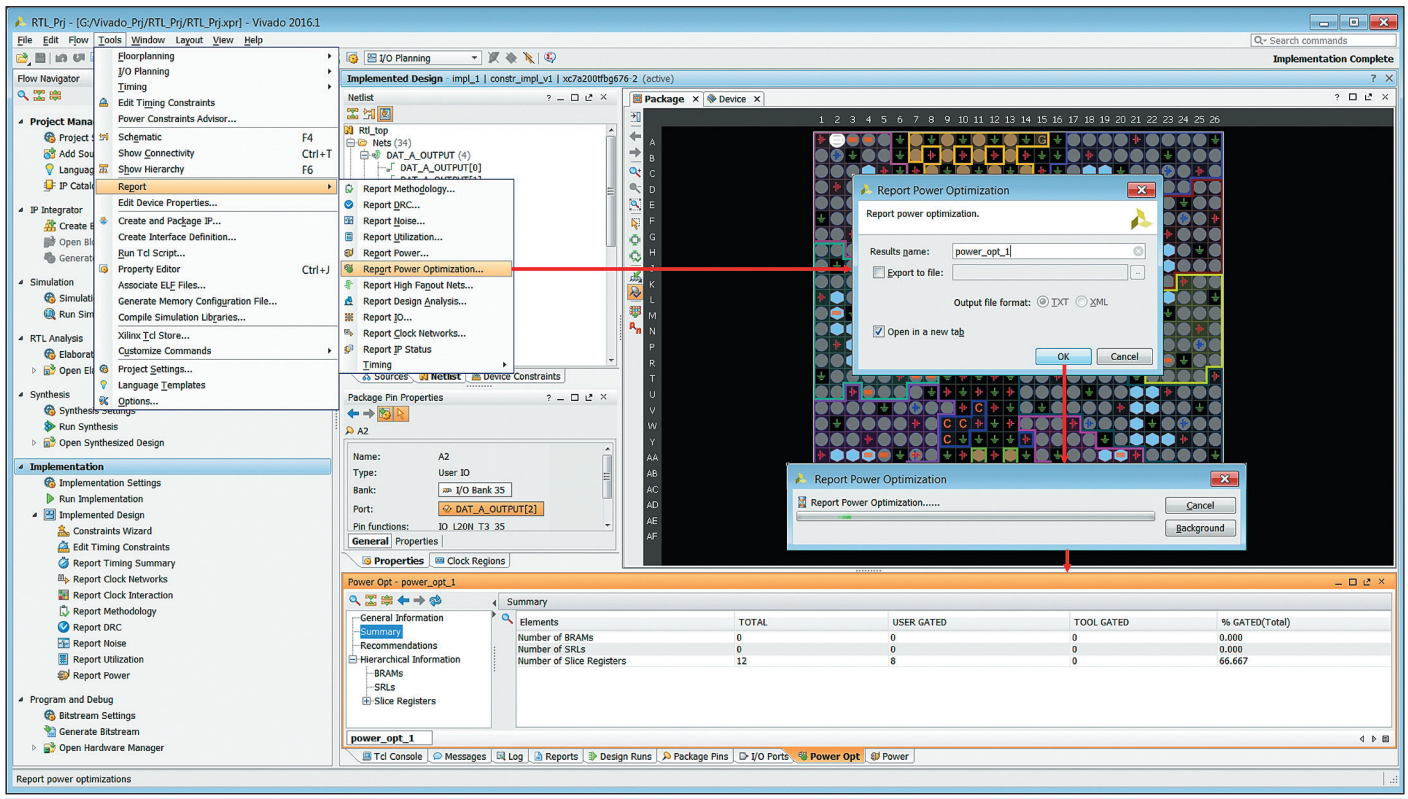

Рис. 12. Формирование отчета, содержащего рекомендации по оптимизации потребляемой мощности разрабатываемого устройства

этой панели. На экране появится информационная панель с заголовком Report Power Optimization, где отображаются краткие сведения о ходе генерации отчета. По завершении формирования рекомендаций по оптимизации потребляемой мощности разрабатываемого устройства в консольной области основного окна интегрированной среды разработки Vivado IDE открывается новая вкладка с названием *Power Opt — <название_отчета>*, чей вид представлен на рис. 12.

Открывшаяся вкладка содержит сформированный отчет, имеющий четыре раздела с названиями General Information, Summary, Recommendations и Hierarchical Information. В разделе General Information приведена общая информация о разрабатываемом проекте, типе кристалла программируемой логики или расширяемой процессорной платформы, используемого для его реализации, версии САПР серии Xilinx Vivado HLx Design Suite, в среде которой сгенерирован отчет, дате и времени создания рассматриваемого документа. Здесь же приведен полный формат предназначенной для генерации отчета команды языка TCL со всеми установленными параметрами. Раздел Summary информирует о количестве модулей блочной памяти, сдвиговых регистров SRL и триггеров конфигурируемых логических блоков, задействованных для реализации выполняемого проекта, которые предоставляют возможности дополнительной оптимизации энергопотребления разрабатываемого устройства. В разделе Recommendations предложены рекомендации по дополнительному снижению уровня потребляемой мощности. Раздел Hierarchical Information включает несколько подразделов с подробными сведениями о каждом модуле блочной памяти, сдвиговом регистре SRL и триггере, применяемом для реализации компонентов различных уровней иерархии реализуемого проекта, которые рассматриваются в процессе оптимизации энергопотребления разрабатываемого устройства.

Дополнительные возможности снижения уровня потребляемой мощности предлагает инструмент Power Constraint Advisor, также предусмотренный в составе средств проектирования серии Xilinx Vivado HLx Design Suite. Этот инструмент осуществляет анализ интенсивности переключений управляющих сигналов проектируемого устройства (сброса, установки, разрешения). В результате проведенного анализа разработчик получает информацию о цепях управления, чьи сигналы находятся в неактивном состоянии чрезмерно длительные интервалы времени, что в ряде случаев приводит к повышению энергопотребления. Указанный инструмент позволяет устранить такие потенциальные причины повышения потребляемой мощности разрабатываемого устройства, добавив необходимые ограничения в реализуемый проект.

Для того чтобы воспользоваться средствами анализа переключений управляющих сигналов проектируемого устройства, следует во всплывающем меню **Tools** основного окна интегрированной среды разработки Vivado IDE выбрать строку Power Constraint Advisor. На экран будет выведена одноименная диалоговая панель, как показано на рис. 13.

Открывшаяся диалоговая панель содержит сведения о цепях управления разрабатываемого устройства, характер поведения сигналов которых может приводить к росту уровня энергопотребления. Эти сведения приведены в форме таблицы; в ней семь колонок с названиями Net, Confidence, Fanout, Fanout Type, Polarity, Static Probability и Toggle Rate. В колонке Net перечислены полные идентификаторы цепей управления с учетом иерархической структуры проекта. Столбец Confidence содержит информацию об уровне достоверности оценок переключения соответствующих управляющих сигналов. В колонке Fanout представлены сведения о количестве разветвлений управляющих цепей. Ячейки столбца Fanout Type информируют о типе управляющих сигналов соответствующих цепей. Колонка Polarity отображает сведения об активном уровне сигналов всех перечисленных цепей управления. В ячейках столбца Static Probability указаны оценки вероятности статического состояния сигналов управления. Ячейки колонки Toggle Rate содержат значения оценки интенсивности переключений сигналов соответствующих цепей управления. При изучении информации, представленной в рассмотренной таблице, следует в первую очередь обратить особое внимание на цепи управления с большим количеством разветвлений и низким уровнем

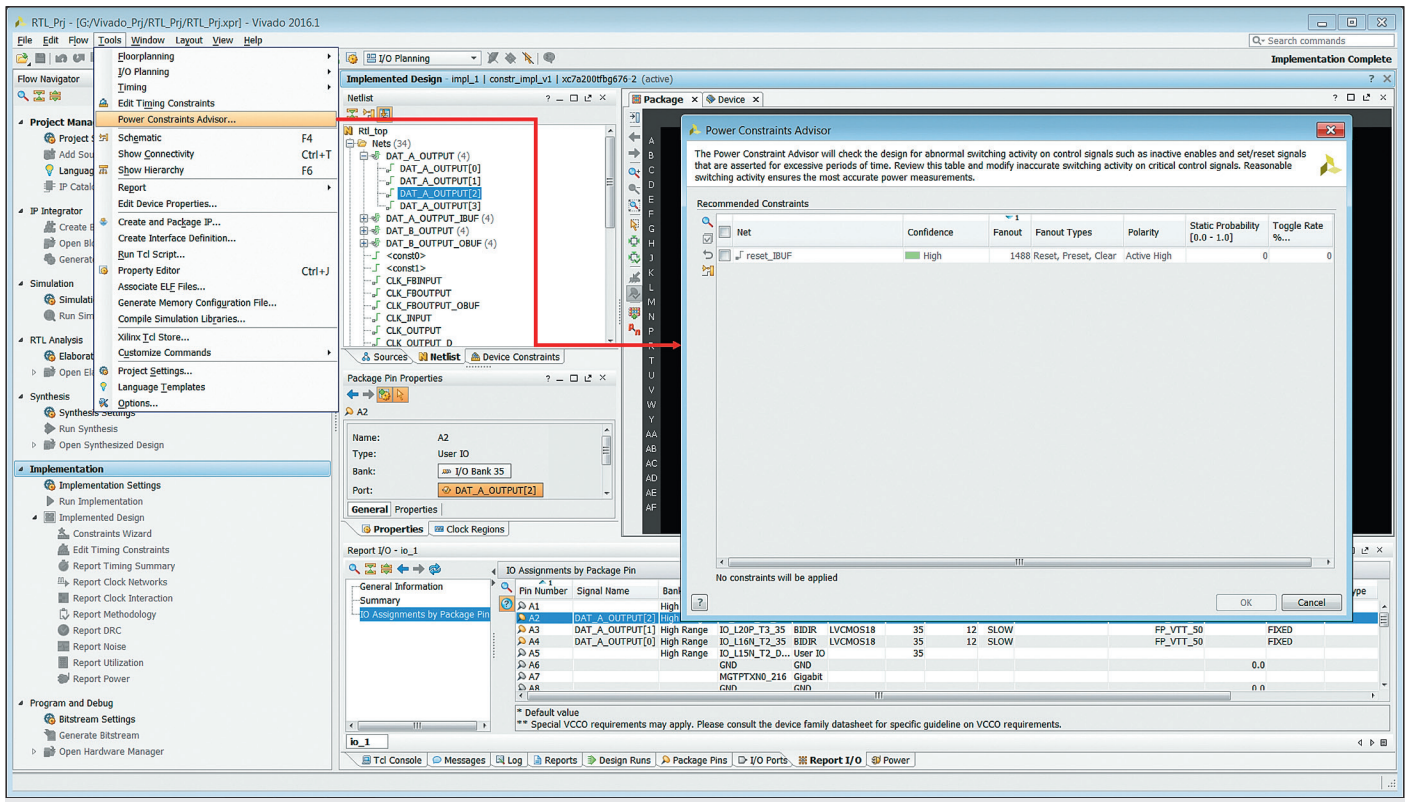

Рис. 13. Активизация средств анализа переключений управляющих сигналов проектируемого устройства Power Constraint Advisor

достоверности приведенных оценок. Длительное неактивное состояние сигналов таких цепей может вызывать существенный рост потребляемой мощности проектируемого устройства.

Для коррекции значений, представленных в ячейках колонок Static Probability и Toggle Rate, необходимо прежде всего установить в положение «Включено» индикаторы состояния в соответствующих строках рассмотренной таблицы. Указанные ячейки таблицы преобразуются в поля редактирования значений нужных показателей, активизация которых выполняется двойным щелчком левой кнопкой мыши при расположении курсора в этих ячейках. Отметив

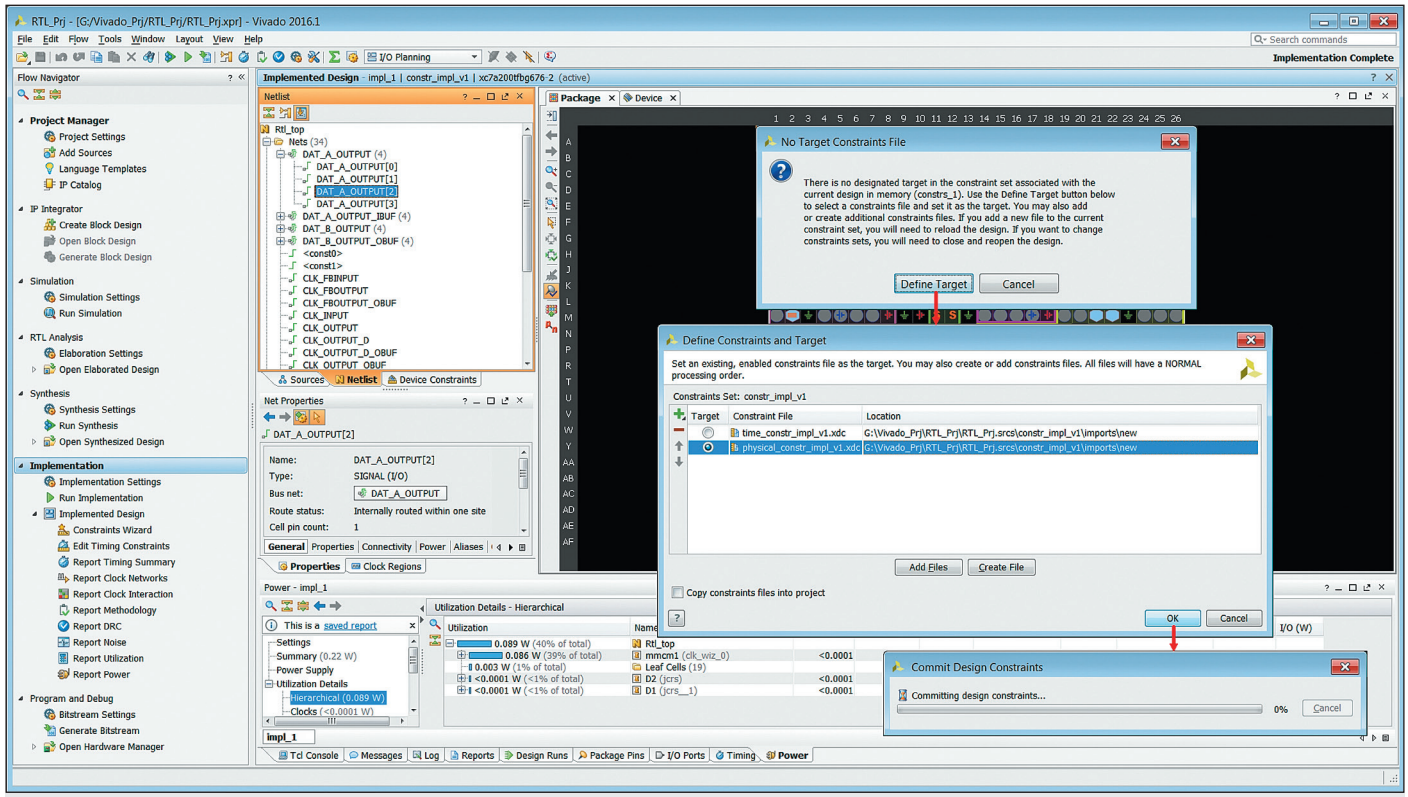

Рис. 14. Запись выражений ограничений, сгенерированных средствами Power Constraint Advisor, в выбранный модуль XDC

с помощью клавиатуры все требуемые значения, следует подтвердить их нажатием клавиши **OK** в нижней части диалоговой панели Power Constraint Advisor, затем будут сгенерированы и записаны в заданный файл XDC выражения ограничений. Если перед запуском средств Power Constraint Advisor не определен целевой модуль XDC, предназначенный для сохранения формируемых ограничений проекта, то на экране появляется диалоговая панель с заголовком No Target Constraints File, содержащая предупреждение. Для определения требуемого файла временных и физических ограничений можно воспользоваться клавишей **Define Target**, расположенной в нижней части этой диалоговой панели. После нажатия клавиши открывается диалоговая панель с заголовком Define Constraints and Target (рис. 14).

Открывшаяся диалоговая панель позволяет выбрать один из предлагаемых модулей XDC или создать новый файл для сохранения выражений ограничений, сгенерированных средствами Power Constraint Advisor. Чтобы указать уже существующий модуль XDC, следует зафиксировать в нажатом состоянии кнопку, расположенную в ячейке Target соответствующей строки таблицы, представленной в диалоговой панели Define Constraints and Target. Для создания нового файла ограничений надо воспользоваться клавишей **Create File**, которая находится под этой таблицей. Выбрав файл ограничений, необходимо нажать клавишу **OK**, расположенную в нижней части диалоговой панели Define Constraints and Target. На экране появляется информационная панель с заголовком Commit Design Constraints, в ней содержится сообщение о записи сформированных выражений ограничений в выбранный файл XDC. Впоследствии рекомендуется повторить этап реализации проектируемого устройства в кристалле программируемой логики или расширяемой процессорной платформы с учетом внесенных ограничений, а также процедуру анализа энергопотребления.

Продолжение следует

Литература

1. Зотов В. Проектирование цифровых устройств на основе ПЛИС фирмы Xilinx в САПР WebPack ISE. М.: Горячая линия – Телеком, 2003.
2. Зотов В. Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы Xilinx. М.: Горячая линия – Телеком, 2006.
3. Зотов В. Проектирование встраиваемых микропроцессорных систем на базе расширяемых процессорных платформ семейства Zynq-7000 AP SoC в САПР Xilinx ISE Design Suite // Компоненты и технологии. 2014. № 4–12. 2015. № 1.
4. Зотов В. Моделирование цифровых устройств, проектируемых на основе ПЛИС фирмы Xilinx, средствами ISIM в САПР ISE Design Suite // Компоненты и технологии. 2013. № 2–3.
5. Зотов В. Средства автоматизированного проектирования и этапы разработки встраиваемых микропроцессорных систем на базе расширяемых процессорных платформ семейства Zynq-7000 AP SoC // Компоненты и технологии. 2014. № 2–3.
6. Зотов В. Особенности архитектуры нового поколения ПЛИС с архитектурой FPGA фирмы Xilinx // Компоненты и технологии. 2010. № 12.
7. Зотов В. Расширение семейства программируемых систем на кристалле Zynq-7000 AP SoC // Компоненты и технологии. 2013. № 12. 2014. № 1.
8. Vivado Design Suite User Guide. Release Notes, Installation and Licensing. Xilinx, 2016.
9. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 1 // Компоненты и технологии. 2016. № 7.
10. Зотов В. Конвертирование проектов цифровых устройств, разрабатываемых на основе ПЛИС и полностью программируемых систем на кристалле фирмы Xilinx в среде ISE Design Suite, в формат САПР Vivado Design Suite // Компоненты и технологии. 2013. № 8–10.
11. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 2 // Компоненты и технологии. 2016. № 8.
12. Зотов В. Разработка VHDL-описаний цифровых устройств, проектируемых на основе ПЛИС фирмы Xilinx, с использованием шаблонов САПР ISE Design Suite // Компоненты и технологии. 2010. № 2–12. 2011. № 1–12. 2012. № 1–12.
13. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 3 // Компоненты и технологии. 2016. № 9.
14. Vivado Design Suite User Guide. Using Constraints. Xilinx, 2016.
15. Vivado Design Suite Properties Reference Guide. Xilinx, 2016.
16. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 4 // Компоненты и технологии. 2016. № 10.
17. Зотов В. Проектирование узлов синхронизации цифровых устройств и встраиваемых микропроцессорных систем, реализуемых на базе ПЛИС фирмы Xilinx серий Artix-7, Kintex-7 и Virtex-7. Часть 1 // Компоненты и технологии. 2016. № 3.
18. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 5 // Компоненты и технологии. 2016. № 11.
19. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 6 // Компоненты и технологии. 2016. № 12.
20. Зотов В. ModelSim — система HDL-моделирования цифровых устройств // Компоненты и технологии. 2002. № 6.
21. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 7 // Компоненты и технологии. 2017. № 1.
22. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 8 // Компоненты и технологии. 2017. № 2.
23. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 9 // Компоненты и технологии. 2017. № 3.
24. UltraFast Design Methodology Quick Reference Guide. Xilinx, 2016.
25. UltraFast Design Methodology Guide for the Vivado Design Suite. Xilinx, 2016.
26. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 10 // Компоненты и технологии. 2017. № 4.
27. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 11 // Компоненты и технологии. 2017. № 5.
28. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 12 //Компоненты и технологии. 2017. № 6.
29. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 13 // Компоненты и технологии. 2017. № 7.
30. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 14 // Компоненты и технологии. 2017. № 8.
31. Vivado Design Suite User Guide: Synthesis. Xilinx, 2016.
32. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 15 // Компоненты и технологии. 2017. № 9.
33. Large FPGA Methodology Guide Including Stacked Silicon Interconnect (SSI) Technology. Xilinx, 2012.
34. Santarini M. Stacked & Loaded: Xilinx SSI, 28-Gbps I/O Yield Amazing FPGAs // Xcell Journal. 2011. № 74.
35. Santarini M. Xilinx Ships World's Highest-Capacity FPGA Using SSI Technology // Xcell Journal. 2011. № 77.
36. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 16 // Компоненты и технологии. 2017. № 10.
37. Зотов В. Проектирование цифровых устройств на базе ПЛИС фирмы Xilinx в САПР серии Vivado HLx Design Suite. Часть 17 // Компоненты и технологии. 2017. № 12.