

Использование IP Integrator в САПР Vivado для ПЛИС серии 7 и UltraScale

Илья ТАРАСОВ,
д. т. н.
ilya_e_tarasov@mail.ru

В течение последних лет производители ПЛИС уделяют много внимания повышению эффективности инструментов проектирования при работе с FPGA большой логической емкости. Одним из таких инструментов, который предназначен для представления проекта в виде блок-схем, является IP Integrator. Он добавлен в САПР Vivado 2013.3, как и поддерживаемый им блочный дизайн.

Введение

В 2012 году компания Xilinx представила новое поколение САПР Vivado. В качестве обоснования актуальности этой САПР было указано то, что постоянный рост объемов ПЛИС с архитектурой FPGA привел к необходимости качественного изменения порядка проектирования в этой области микроэлектроники. Обновление в Vivado коснулось двух основных аспектов — основного маршрута САПР (включая синтез, автоматическое размещение и трассировку) и порядка описания схем.

Что касается собственно маршрута проектирования, то в материалах Xilinx отмечалось, что предыдущее поколение синтезатора XST (Xilinx Synthesis Technology) и процесса implementation имели предел приблизительно в 2 млн логических ячеек, при достижении которого качество трассировки заметно ухудшалось. Причиной этого ухудшения стало, прежде всего, то, что создание XST относится к версиям САПР 10-летней давности, которые были ориентированы на соответствующие, гораздо меньшие объемы FPGA, и, очевидно, решали те задачи синтеза и трассировки, которые были актуальны на тот момент.

За те 10 лет, в течение которых маршрут XST функционировал в качестве основного, объемы FPGA увеличились с тысяч ячеек до миллионов. В принципе это является неплохим показателем масштабируемости технологии, однако с выходом серии 7, максимальный объем которой составляет 2 млн ячеек, остро встал вопрос о замене алгоритмов синтеза и трассировки. Решением этой проблемы стал новый маршрут проектирования Vivado Synthesis Technology, который, по некоторым оценкам Xilinx, масштабируется по меньшей мере до 20 млн ячеек. Для семейств, следующих за серией 7, Vivado будет единственным доступным маршрутом проектирования: САПР ISE не будет более пригоден для этой цели.

Увеличение объемов FPGA влечет за собой и другую проблему — необходимость повышения эффективности труда разработчика, для которого становится сложным не только обеспечить высокие характеристики созданного устройства, но и хотя бы заполнить кристалл актуальным проектом. Графический ввод схемы просто не позволяет сделать это за разумное время. Поэтому схемотехническое представление из САПР Vivado оказалось исключено, а в качестве компонентов выступают RTL-описания (то есть тексты на языках VHDL, Verilog и System Verilog) и различные компоненты системного уровня, например процессорные системы, создаваемые с помощью САПР EDK, или подсистемы цифровой обработки сигналов, проектируемые с помощью System Generator for DSP. Недавно к списку инструментов высокого уровня добавилась САПР Vivado HLS, предусматривающая описание проекта с использованием синтаксиса языков C/C++/SystemC с последующим преобразованием описанных алгоритмов в RTL-описание. Эти инструменты позволяют в более короткие сроки и с большей наглядностью описать устройства, которые потребовали бы на несколько порядков большей трудоемкости при попытке представления их в виде принципиальной электрической схемы на уровне логических вентилях и регистров.

В то же время графическое представление проектов в ряде случаев имеет большую наглядность, позволяя анализировать взаимодействие крупных блоков, их интерфейсы, потоки данных и пр. Анализ предпочтений разработчиков, сделанный на базе курсов Xilinx, которые проходят в учебном центре компании КТЦ «Инлайнгруп», показал, что для САПР ISE многие инженеры считают возможной (а некоторые — и предпочтительной) структуру проекта, при которой верхним уровнем является именно схема. Однако на такой схеме размещаются не отдельные

низкоуровневые компоненты, а крупные блоки, содержащие тысячи или десятки тысяч логических ячеек. Графическое представление гораздо удобнее для совместного анализа работы устройства с участием специалистов по смежным направлениям или специалистов в предметной области проекта, которые могут не владеть языками описания аппаратуры и, таким образом, не смогут проанализировать структуру проекта на базе HDL-представления верхнего уровня.

Так или иначе в САПР Vivado компания Xilinx отказалась от графического ввода схемы. Взамен этого предлагается инструмент под названием IP Integrator, который призван обеспечить графическое представление проекта, но уже не для низкоуровневых компонентов, а для IP-ядер (как и следует из его названия). Актуальность такого способа ввода становится понятной, если учесть эффективность повторного использования кода (англ. термин *code reuse*), который при программировании выступает в виде готовых библиотек, а при разработке проектов на базе ПЛИС — в виде готовых функциональных компонентов — IP-ядер.

Назначение и порядок работы с IP Integrator

Инструмент IP Integrator был добавлен в САПР Vivado 2013.3. С ним связано понятие блочного проекта (block design), который может выступать в качестве элемента иерархии наряду с модулями на языках VHDL и Verilog. Порядок работы с IP Integrator удобнее рассмотреть на практическом примере.

Создав новый проект на базе платы Zedboard, можно выбрать для этой платы новый блочный проект, как показано на рис. 1. На этом рисунке можно видеть, что на панели **Flow Navigator** появилась новая группа под названием **IP Integrator**. При выборе в этой группе пункта **Create Block Design** на-

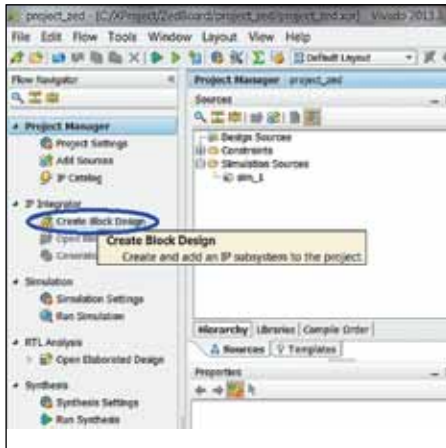


Рис. 1. Группа IP Integrator и создание блочного проекта в САПР Vivado 2013.3

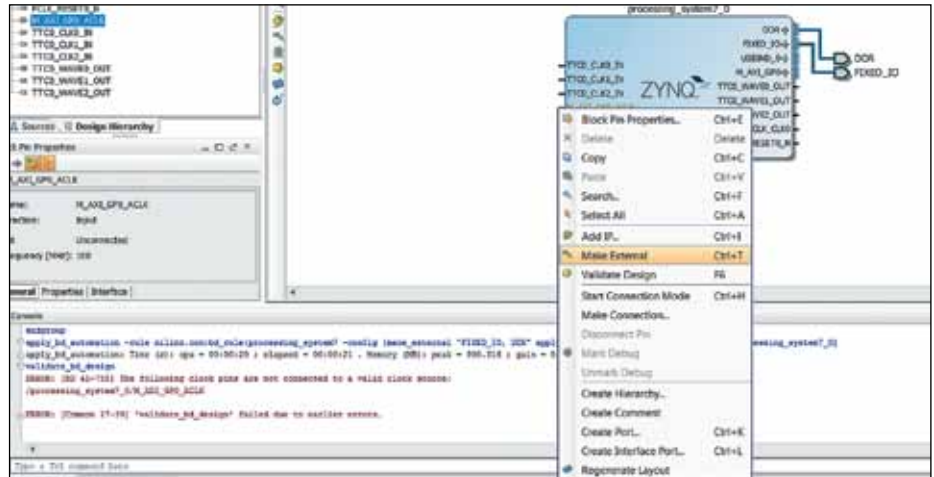


Рис. 4. Добавление внешнего тактового сигнала в проект

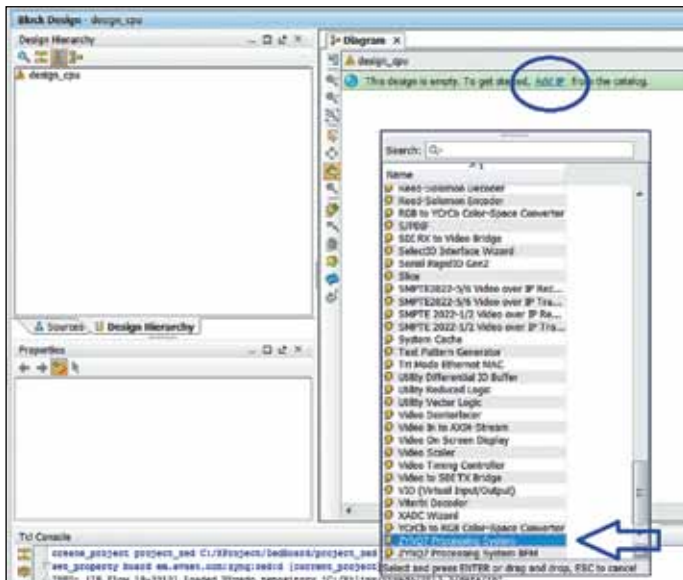


Рис. 2. Добавление IP-ядра к блочному проекту в САПР Vivado



Рис. 3. Представление процессорной подсистемы на базе ARM в виде блочного проекта в САПР Vivado

чинается работа «мастера» создания новой подсистемы на основе IP-ядра. В данном примере для подсистемы выбрано имя `design_cpu`.

Добавить IP-ядро к блочному проекту можно с помощью элемента управления, оформленного в виде гиперссылки, как показано на рис. 2. Нажатие на элемент **Add IP** открывает список доступных для выбранной ПЛИС IP-ядер. После выбора компонента

Zynq7 Processing System и завершения работы генератора будет сформировано графическое представление процессорной подсистемы на основе ARM, входящей в состав ПЛИС Zynq. На рис. 3 показан интерфейс САПР Vivado после установки такой подсистемы. Для установленного ядра доступна автоматическая настройка его параметров, что видно по сообщению “Designer Assistance available” и гиперссылке, включающей такую автонастройку.

В результате работы блока автонастройки к процессорной подсистеме ARM будут подключены внешняя память DDR3 и периферийные устройства, контроллеры которых имеются на плате Zedboard. Несомненным удобством является наличие функции **Validate Design** (назначенной также на клавишу **F6**), однако запуск проверки для автоматически настроенной процессорной подсистемы показывает ошибку: тактовый сигнал при автонастройке на систему не подается. На рис. 4 показана эта ситуация вместе со способом ее исправления. Щелчок правой кнопкой мыши по выводу `M_AXI_GPO_ACLK` открывает контекстное меню, в котором есть пункт **Make External**.

Далее для созданного проекта необходимо запустить процесс **Generate Block Design** (он также находится в группе **IP Integrator**). При этом создаются необходимые файлы для синтеза и моделирования созданной подсистемы.

Разработчики, знакомые с САПР EDK, могли заметить, что пункт **Make External** встречается именно в этой САПР. Однако это далеко не единственная функциональность, перенесенная из EDK (а точнее, из программы Xilinx Platform Studio, с помощью которой выполняется конфигурирование процессорных подсистем). Двойной щелчок на графическом изображении Zynq открывает диалоговое окно конфигурирования непосредственно в САПР Vivado (рис. 5). На этом рисунке показана также часть интерфейса Vivado, чтобы подчеркнуть, что настройка процессорной подсистемы, для которой ранее требовался запуск Xilinx Platform Studio, теперь может быть выполнена непосредственно в интерфейсе Vivado. Подробное изучение деталей настройки представляется нецелесообразным в рамках данной статьи.

Однако хотелось бы обратить внимание на удобную функцию, упрощающую распределение внешних выводов подсистемы ARM. Дело в том, что при интеграции процессорного ядра ARM в ПЛИС семейства Zynq были приняты определенные компромиссные решения, касающиеся организации внешних выводов. Аппаратные контроллеры периферии, входящие в состав подсистемы ARM, в принципе позволяют подключить множество внешних устройств, задействовав при этом большое число выводов микросхемы. Но это потребовало бы использования для ПЛИС корпусов с большим числом выводов, заметная часть которых относилась бы к встроенной аппаратной периферии ARM, пусть даже и редко используемой. Поэтому при создании ПЛИС Zynq было использовано мультиплексирование сигналов аппаратных контроллеров — Multiplexed IO (MIO).

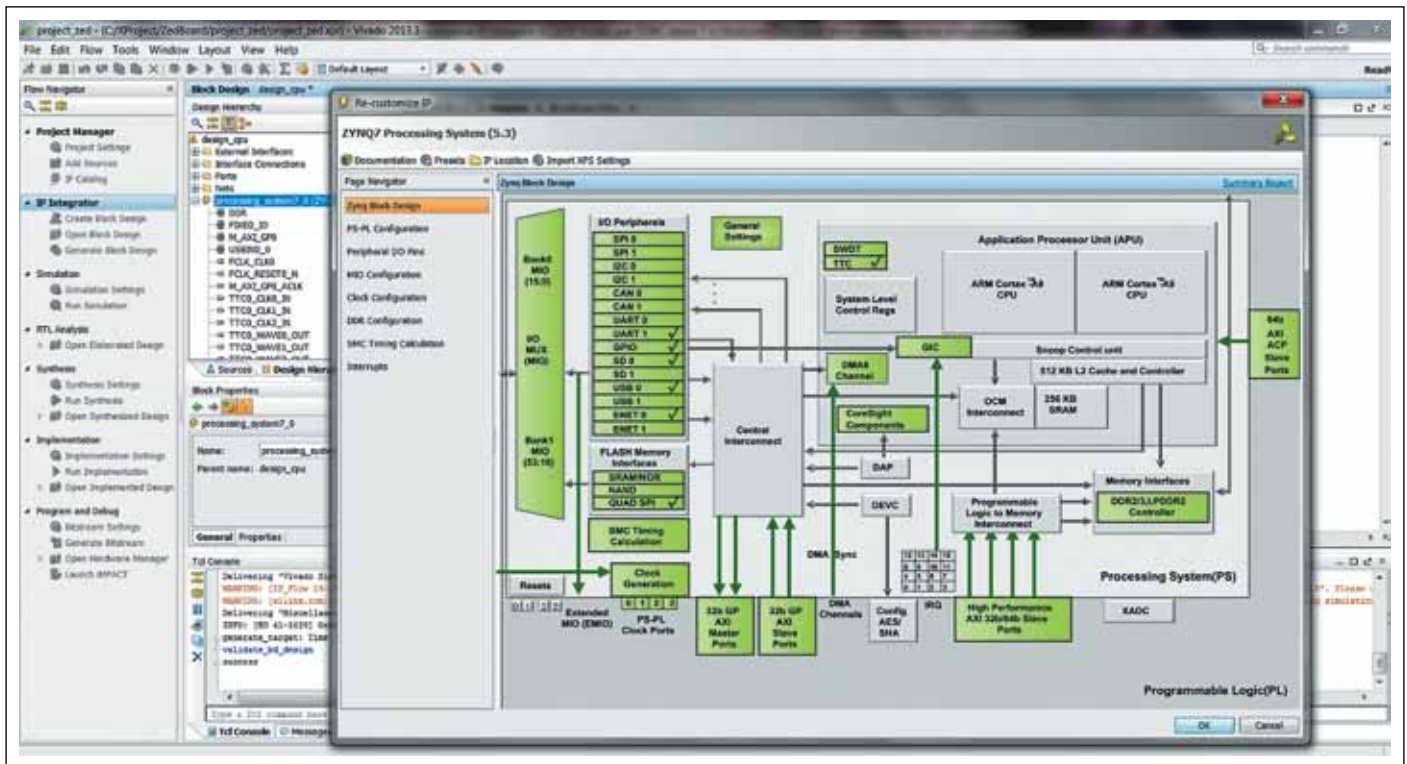


Рис. 5. Окно настройки параметров процессорной подсистемы ARM

В ПЛИС Zynq имеются 54 внешних вывода, принадлежащих подсистеме ARM. Эти выводы недоступны FPGA-части кристалла и могут быть использованы только встроенной периферией ARM. Такого количества сигналов недостаточно для подключения всех устройств, поэтому разработчику необходимо определить, какие именно контроллеры будут использоваться в проекте. Ситуация несколько осложняется тем, что внешние выводы не могут быть назначены произвольно. Каждый контроллер имеет несколько вариантов размещения, что мо-

жет привести к конфликтам, когда даже при формальном наличии свободных выводов используемые периферийные контроллеры не могут быть задействованы одновременно. Это касается, в частности, аппаратных контроллеров Ethernet MAC и USB. В каждой ПЛИС Zynq есть по два таких контроллера, однако одновременно на MIO могут быть выведены только два из четырех устройств (то есть или два USB, или два EMAC, или один USB + один EMAC). Такую особенность организации периферийных контроллеров необходимо заранее учитывать при

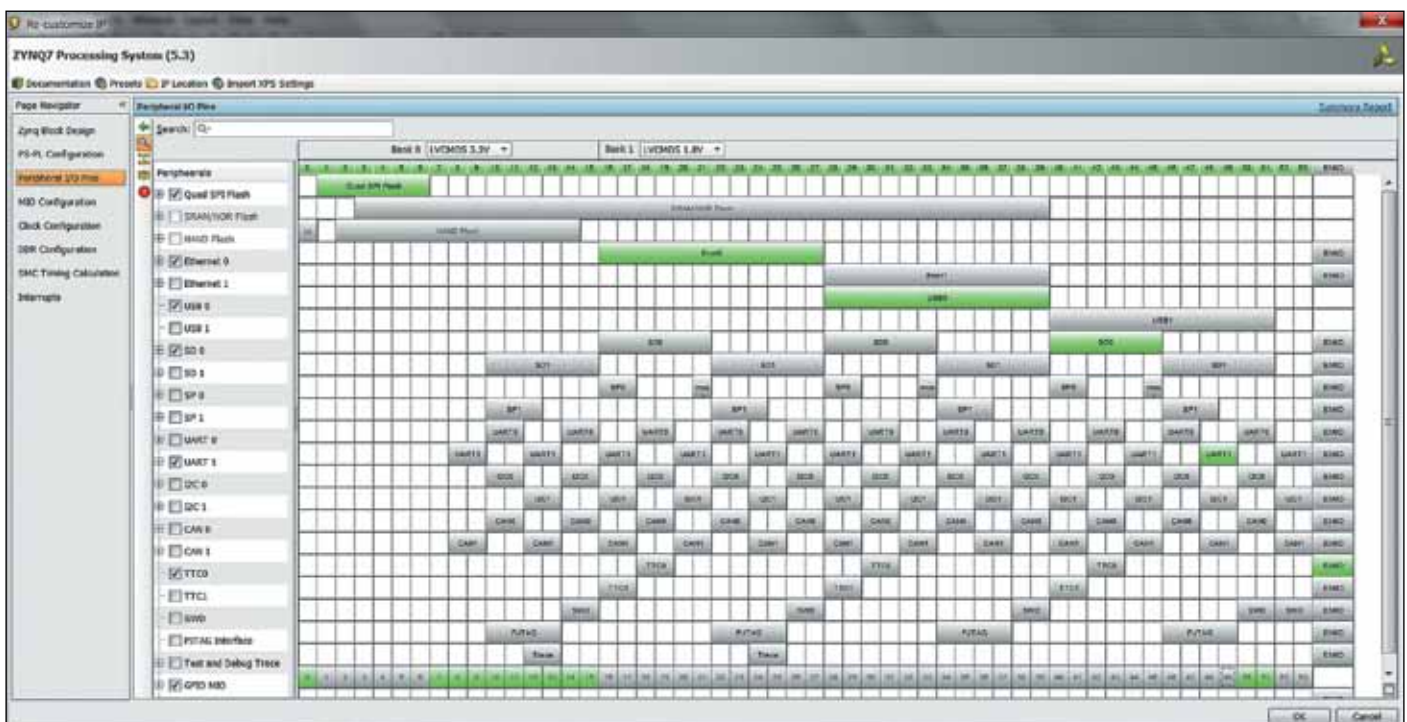


Рис. 6. Графическое представление распределения внешних выводов периферийных устройств процессорной подсистемы ARM

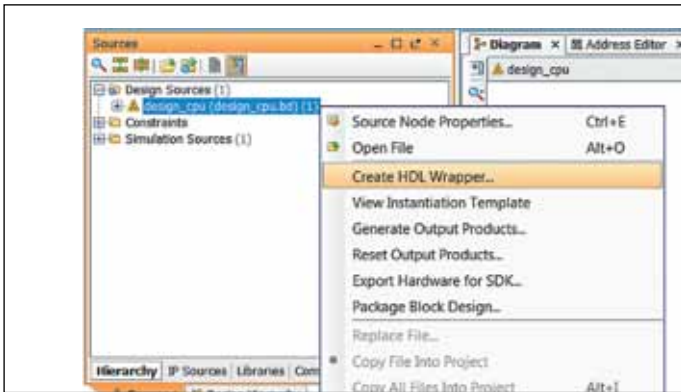


Рис. 7. Создание «обертки» для блочного проекта

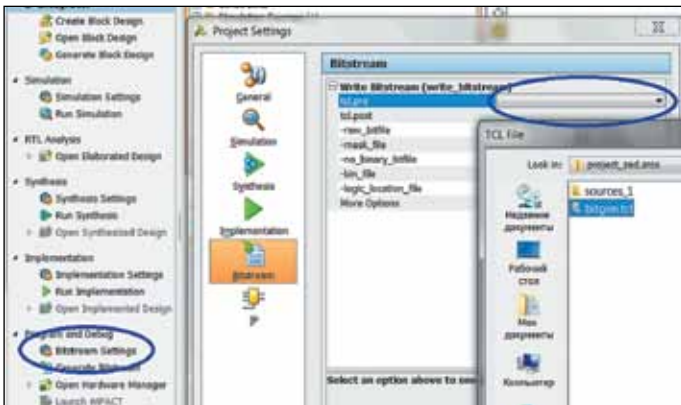


Рис. 8. Настройка сценария, исправляющего ошибку в проверке параметров конфигурации

планировании проекта на базе ПЛИС Zynq, и распределение выводов может быть неочевидным. Однако на рис. 6 эта ситуация представлена наглядно: нетрудно убедиться, что все активированные периферийные устройства не перекрывают друг друга.

Блочный проект не может выступать в качестве модуля верхнего уровня. Поэтому для него необходимо создать «обертку» (wrapper), что также автоматизировано в Vivado. Ее создание выполняется из вкладки Sources (рис. 7).

После выполнения описанных действий можно приступить к запуску процессорной системы ARM на плате Zedboard. В контекстном меню на рис. 7 виден также пункт **Export Hardware for SDK**, который выгружает параметры аппаратной платформы в среду разработки программного обеспечения SDK, выполненную на базе IDE Eclipse. Это позволяет сразу после загрузки конфигурации в ПЛИС приступить к программированию.

Текущая версия Vivado (2013.3) содержит несущественную недоработку, препятствующую получению конфигурационного файла. Из-за неполного описания проектных ограничений на стадии bitgen формируется ошибка, связанная с отсутствием описания электрического стандарта ввода/вывода для некоторых сигналов. Для решения этой проблемы необходимо создать файл сценария на языке tcl, указав его в качестве «предварительно запускаемого» (*tclpre*) в настройках процесса **Bitstream**. На рис. 8 показано диалоговое окно настройки параметров процесса **Bitstream** и порядок действий для указания размещения требуемого файла. Описание этой проблемы приведено в [1], ее исправление не может быть произведено путем указания ограничений в формате *xdc*.

Сам файл должен быть создан предварительно и иметь следующее содержимое:

```
set_property SEVERITY {Warning} [get_drc_checks NSTD-1]
set_property SEVERITY {Warning} [get_drc_checks UCIO-1]
```

Перспективы применения Vivado для FPGA с архитектурой UltraScale

11 ноября 2013 года компания Xilinx выпустила пресс-релиз [2], в котором сообщила об отгрузке потребителю первой в мире ПЛИС, выполненной с применением 20-нм норм. На момент подготовки материала статьи технические детали не были переданы в свободный доступ, однако речь в пресс-релизе идет о микросхеме, выполненной по новой архитектуре UltraScale. Архитектура призвана развивать преимущества ПЛИС серии 7. Планы освоения технологических процессов представлены на рис. 9.

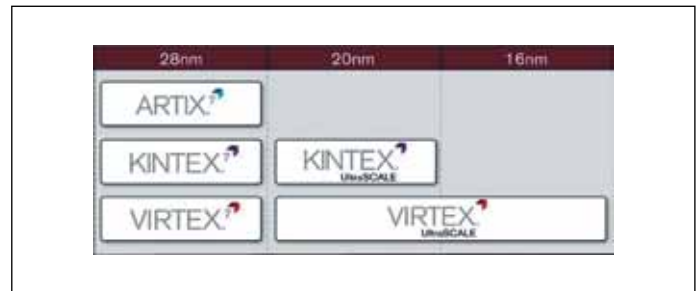


Рис. 9. Планы выпуска семейств ПЛИС с архитектурой UltraScale

Среди преимущественных применений нового поколения ПЛИС упоминаются:

- 400G OTN;
- 4X4 Mixed Mode LTE;
- WCDMA Radio;
- дисплеи 4K2K и 8K;
- суперкомпьютеры и центры обработки данных.

Важно, что архитектура UltraScale не будет поддерживаться в САПР ISE, поэтому для использования этих ПЛИС переход к САПР Vivado становится обязательным для разработчиков. Можно ожидать, что новое поколение микросхем в очередной раз предоставит в их распоряжение увеличенную логическую емкость, лучшие функциональные возможности отдельных блоков и улучшенную производительность САПР. Однако одновременно с этим потребуются освоить соответствующие методики проектирования, которые позволят специалистам ориентироваться во всем богатстве аппаратных ресурсов новых ПЛИС и эффективно их использовать. В очередной раз можно подчеркнуть, что глубокое владение методиками и средствами проектирования становится все более важным в такой высокотехнологичной области, как программируемая цифровая микроэлектроника.

Заключение

Показанный в статье пример демонстрирует использование нового инструмента, включенного в состав САПР Vivado. Применение блочного представления для крупных IP-ядер позволяет повысить наглядность процесса проектирования, автоматизировав его в дополнительных аспектах. Проектирование на верхнем уровне существенно облегчается, в том числе для специалистов в области разработки программного обеспечения (что иллюстрирует пример с настройкой подсистемы ARM) или для разработчиков систем цифровой обработки сигналов.

Литература

1. Xilinx Answer Record# 56354 — <http://www.xilinx.com/support/answers/56354.htm>
2. Xilinx Ships Industry's First 20nm All Programmable Product — <http://press.xilinx.com/2013-11-11-Xilinx-Ships-Industrys-First-20nm-All-Programmable-Product>