

Быстрая разработка устройств ЦОС в системе Xilinx System Generator

Александр Литвинов (г. Воронеж)

Статья содержит краткое описание программного пакета Xilinx System Generator и показывает преимущества модельно-ориентированного проектирования устройств ЦОС на ПЛИС в данной среде.

ВВЕДЕНИЕ

Первую версию пакета проектирования System Generator фирма Xilinx выпустила ещё в 2000 г. Несмотря на это, пакет до сих пор не получил широкого распространения среди разработчиков. Причём тенденция одинакова и в России, где сейчас работает автор, и, например, в Samsung Electronics, где автор работал ранее. Причин тому много: часть инженеров принципиально не доверяет разработке с помощью IP-блоков, часть работает на языке C-VHDL, другие не освоили Matlab Simulink на необходимом уровне.

Автор уже более 10 лет работает в данной системе и может сказать о ней только хорошее. Так, языком VHDL ему приходилось пользоваться лишь в редких случаях при использовании специфических элементов ПЛИС Xilinx.

Требования к скорости проектирования устройств ЦОС возросли. Если пять лет назад в проектах на ПЛИС не-

обходимо было моделировать 10 – 20 фильтров, то современные системы стали многоканальными и широкополосными, на несколько порядков возросли их объём и сложность.

Выход один – переходить на системный уровень проектирования на уровне больших IP-блоков с одновременным уменьшением времени моделирования. Среда Matlab Simulink позволяет быстро моделировать большие телекоммуникационные проекты (IEEE 802.11, DVB-T, MIMO- и OFDM-демодуляторы). Пакет System Generator в той же среде решает вопрос быстрой реализации устройств ЦОС на ПЛИС Xilinx.

Сведения о ПЛИС для ЦОС

ПЛИС давно используются для построения высокопроизводительных систем ЦОС. Два основных довода в пользу применения ПЛИС:

- архитектура ПЛИС позволяет реализовать параллельную обработку;

- разработчик может оптимизировать проект по производительности и стоимости.

Приведём четыре основных преимущества данной методологии:

- 1) конфигурируемая матрица ПЛИС позволяет создавать параллельные устройства (см. рис. 1);
- 2) возможность выбора между алгоритмами и ресурсами (см. рис. 2);
- 3) низкая стоимость современных приборов. Так, например, кристалл XC6SLX150T (семейство Spartan 6) содержит:
 - 180 умножителей 18×18,
 - 256 блоков памяти по 18 Кбит,
 - 184 000 логических ячеек и столько же триггеров,
 - четыре контроллера памяти,
 - восемь блоков трансиверов со скоростью до 3,125 Гбит/с.
 При этом он обеспечивает производительность более 30 млрд операций умножения-сложения в секунду и продаётся за 180 долл. США.
- 4) быстрая реализация с верхнего уровня (MATLAB) до вентиляльного представления (см. рис. 3).

БИБЛИОТЕКИ SYSTEM GENERATOR

Приведённый ниже список библиотек, естественно, далеко не полный [1]. Внутри только одного блока КИХ-фильтра возможно выбрать десятки реализаций. Он может быть – на выбор – и систолическим MAC, и транспонированным MAC, и с распределённой арифметикой (без умножителей), с интерполяцией и децимацией (в том числе, с дробной в пределах частоты дискретизации), с полифазным банком и разными типами коэффициентов. Всё это автоматически приводится к максимальной тактовой частоте, что позволяет многократно использовать умножитель и экономить ресурсы.

Все блоки оптимизированы по реализации, быстродействию и по размещению в кристалле. К блокам автоматически добавляются константы для указания требований по быстродействию при компиляции ПЛИС. Все блоки – параметризованные. Все задержки в бло-

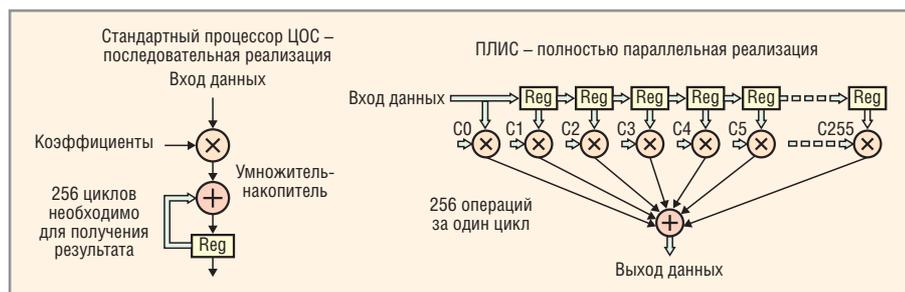


Рис. 1. Параллельная и последовательная реализации

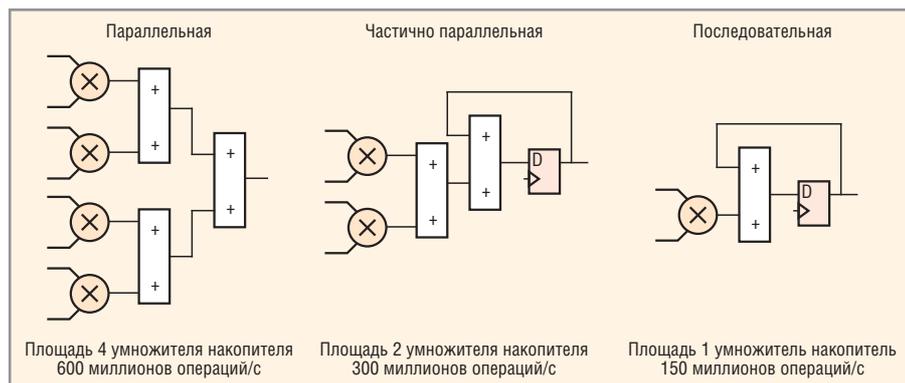


Рис. 2. Возможность выбора между быстродействием и ресурсами



Рис. 3. Методология проектирования устройств ЦОС

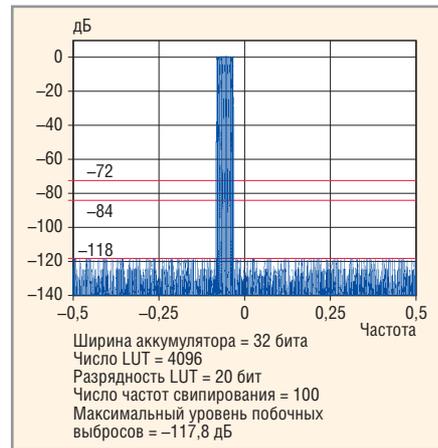


Рис. 4. Пример АЧХ блока цифрового синтезатора частоты фирмы Xilinx (SFDR = 118 дБ) в режиме качания частоты



Рис. 5. Модель верхнего уровня на стандартных блоках Matlab Simulink

как сразу видны при моделировании и совпадают с последующей реализацией в ПЛИС. Пример – результат работы синтезатора частоты Xilinx (см. рис. 4).

- Список доступных IP-блоков:
- фильтры КИХ, фильтры Хогенауэра (СIC), преобразование Фурье (длиной до 65 536 точек, в реальном времени или оптимизированные по ресурсам), ОЗУ, двухпортовое ОЗУ, FIFO, ПЗУ, синтезатор частоты, умножитель, комплексный умножитель, делитель, арктангенс, натуральный логарифм, квадратный корень из суммы квадратов, микроконтроллер PicoBlaze, блок для собственных функций на сокращенном m-языке;
 - долгожданные (начиная с 14 версии ISE) блоки с плавающей запятой:

преобразование Фурье, умножитель, делитель, натуральный логарифм, квадратный корень;

- лицензионные (платные) блоки: перемещитель – деперемещитель, сверточный кодер – декодер, кодер – декодер Рида-Соломона, декодер Виттерби.

Можно добавить в проект и библиотечный блок ChipScore для последующего внутрисхемного анализа. Можно делать вставки на языках и VHDL, и Verilog, и моделировать непосредственно в Matlab, совместно с остальными блоками.

Таблица 1. Сравнение классических методов проектирования ПЛИС с модельно-ориентированным

Традиционные методы	Модельно-ориентированный
Текстовый ввод препятствует быстрой реализации по заданным требованиям	Графический ввод на основе готовых параметрических библиотек
Моделирование неполное и дорогое	Быстрое моделирование сразу во временной и частотной области
Длительные итерации для выбора разрядности с фиксированной точкой	Единая среда моделирования проектов с плавающей и фиксированной точкой
Ручной ввод кода вносит дополнительные ошибки из-за человеческого фактора	Автоматическая генерация VHDL-описания и тестов
При традиционном тестировании ошибки выявляются слишком поздно	Быстрый возврат к верхнему уровню проекта в случае изменения спецификации

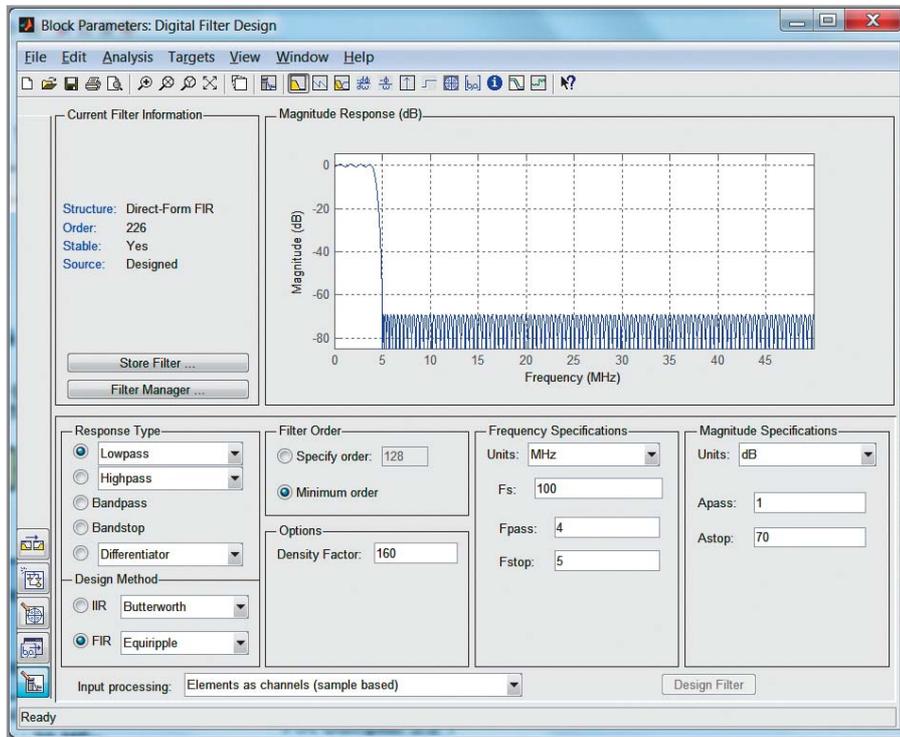


Рис. 6. Окно параметров цифрового фильтра и вид АЧХ

КОРОТКО О ПАКЕТЕ МАТЛАБ

Пакет Matlab фактически стал одним из самых популярных инструментов. Для нас представляют интерес его библиотеки DSP system toolbox и Communication toolbox, содержащие тысячи готовых элементов для ЦОС и телекоммуникаций. Одних только примеров модуляторов, демодуляторов, беспроводных каналов передачи – сотни. Это значительно ускоряет разработку проекта на высоком уровне. Кроме того,

библиотечные блоки могут быть извлечены в виде исходных текстов на языке Си.

МЕТОДЫ ПРОЕКТИРОВАНИЯ

В таблице 1 приведено сравнение классических методов проектирования с модельно-ориентированным. Видно, что подход, реализованный в пакете System Generator, имеет преимущества на каждом этапе проектирования. Поэтому в настоящее вре-

мя многие компании его развивают, и, в том числе, сама компания Mathworks пытается выйти на рынок с новым продуктом – HDL Coder, что говорит о росте интереса к модельно-ориентированным системам проектирования.

ПРОЕКТИРОВАНИЕ В ПАКЕТЕ SYSTEM GENERATOR

Продемонстрируем проектирование на примере блока цифрового переноса частоты вниз. Сначала нарисуем проект в среде Simulink с использованием стандартных блоков пакета Matlab (см. рис. 5). Это – модель верхнего уровня с плавающей точкой. Сигнал передаётся на промежуточной частоте 70 МГц, частота дискретизации 100 МГц, полоса пропускания цифрового фильтра 8 МГц (см. рис. 6). Подавление вне полосы – около 70 дБ. Перенесём сигнал на нулевую частоту с помощью умножителей. Таким способом можно децимировать выходной сигнал и перейти на частоту дискретизации 10 МГц (с комплексными данными). Следовательно, мы всего за несколько минут получили модель верхнего уровня. Время моделирования 10-мс модели составило примерно 1 с.

Перейдём к проектированию с библиотекой System Generator. Открываем Simulink Library Browser и выбираем библиотеку Xilinx Blockset. Нам понадобятся элементы: фильтр (FIR Compiler), синтезатор частоты (DDS Com-

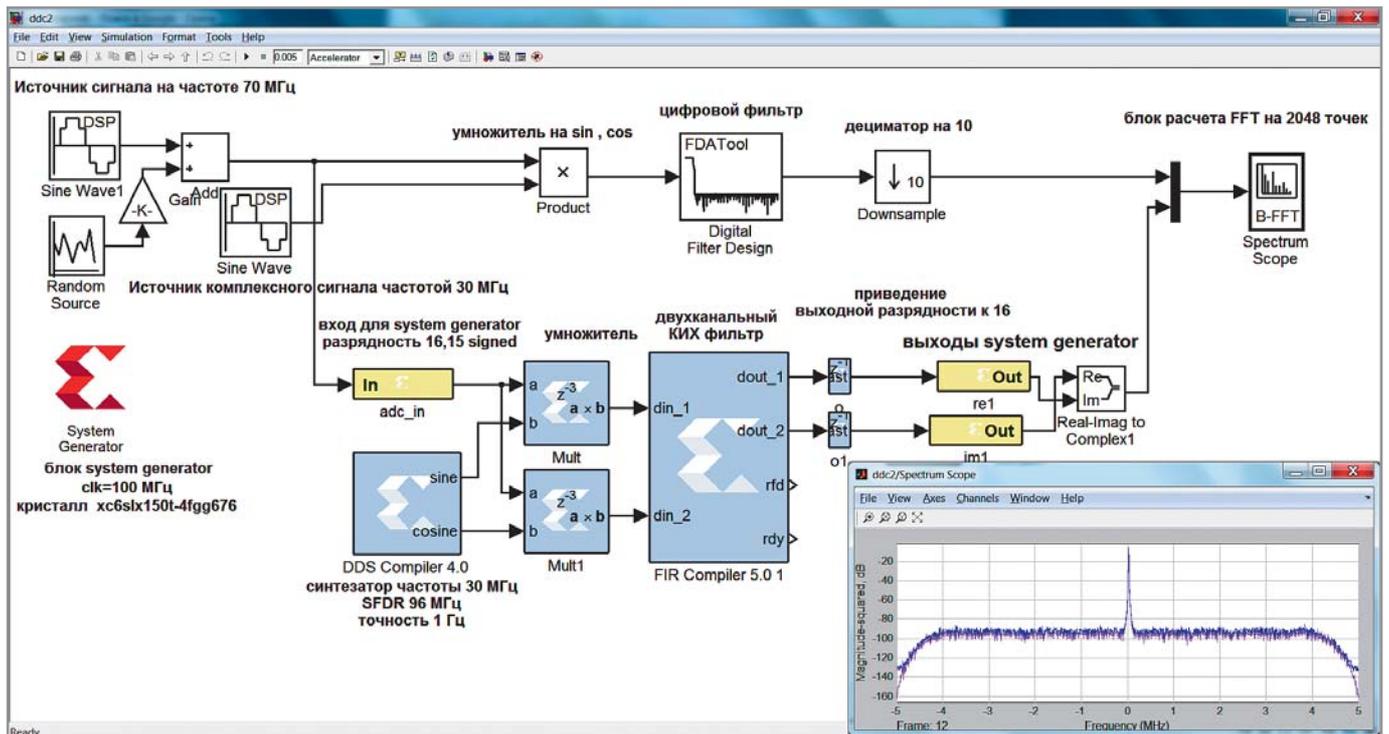


Рис. 7. Модель верхнего уровня и модель в элементах System Generator

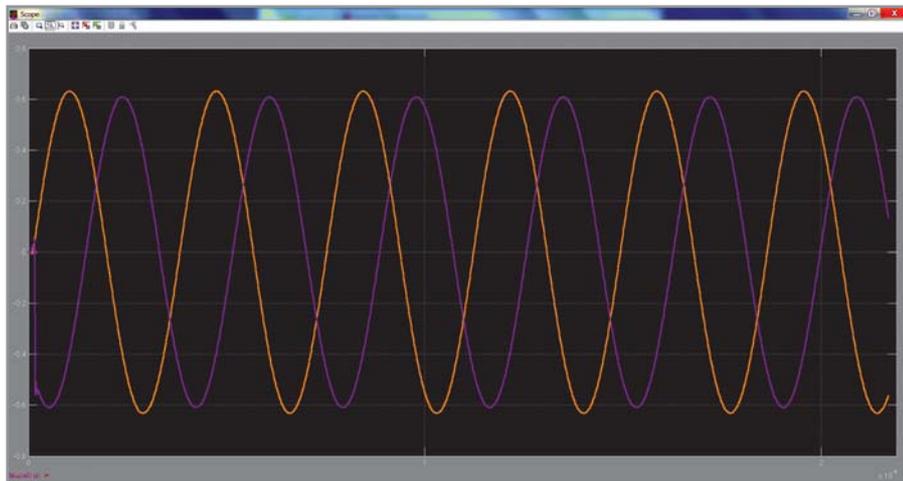


Рис. 8. Вид выходного сигнала во временной области (реальные части с плавающей и фиксированной точкой)

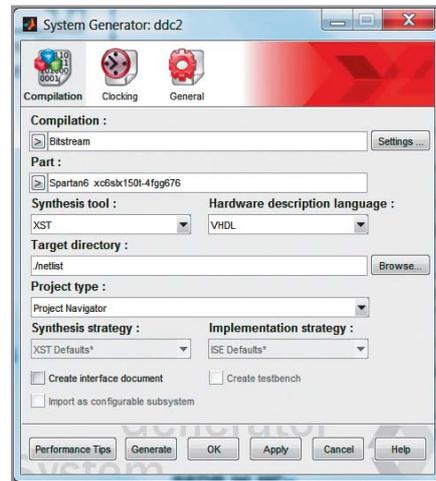


Рис. 9. Вид параметров блока System Generator

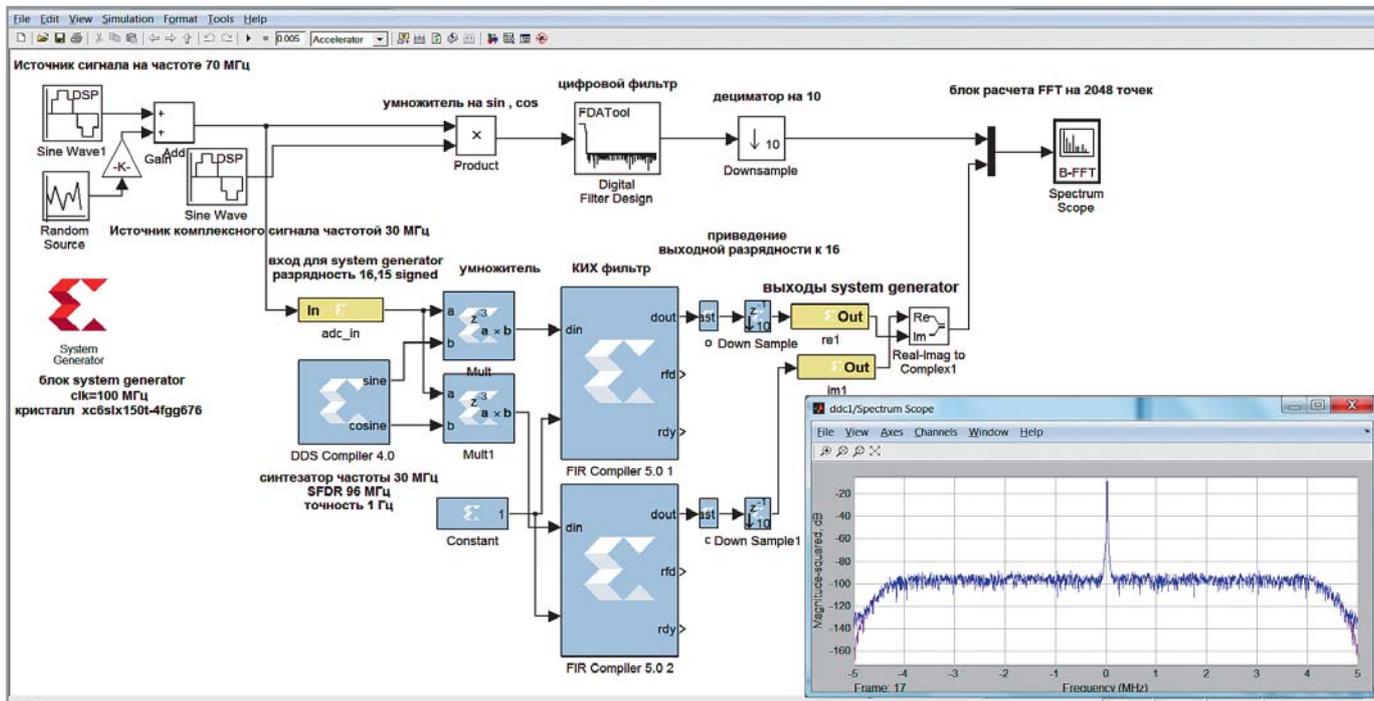


Рис. 10. Модель для оптимизации умножителей

piler), умножители (mult), а также элемент System Generator для указания типа кристалла ПЛИС, параметров компиляции и тактовой частоты, входные и выходные буферы Gate way In и Gate way Out. Все элементы должны быть параметризованы. Рисуем и настраиваем проект (см. рис. 7). Коэффициенты в FIR Compiler экспортируем прямо из блока Digital Filter Design. Для проекта на Xilinx выберем разрядность с фиксированной точкой 16,15 Signed.

Моделируем проект. В блоке Spectrum Score видна разница между проектом с плавающей запятой (сиреневый график) и проектом System Generator с фиксированной точкой (синий график). В любой точке проекта можно посмотреть спектр сигнала или его вид на временной оси. Время моделирования, конечно, увеличилось,

но всё равно оказалось на порядок меньше времени VHDL-симуляции. Сигнал можно проанализировать в любом узле и во временной области (см. рис. 8).

Следует отметить, что моделирование можно ускорить примерно на порядок, используя плату с ПЛИС (Hardware Cosimulation) через интерфейс USB или Ethernet. Данная методика

настолько интересна, что может быть описана в отдельной статье.

Переходим к компиляции проекта в ПЛИС. В блоках Gate way In и Gate way Out назначаем выходы на корпусе ПЛИС. В блоке System Generator (см. рис. 9) указываем тип компиляции Bitstream. Таким образом, получим готовый загрузочный файл для нашей ПЛИС, – достаточно нажать Generate.

Таблица 2. Ресурсы проекта на ПЛИС XC6SLX150T с использованием умножителей

Название ресурса	Использовано в проекте	Всего	В % от ресурса
Количество триггеров	1854	184 304	1
Число программируемых логических таблиц, использованных как логика (LUT)	543	92 152	0,6
Число программируемых логических таблиц, использованных как память (LUT)	1382	21 680	6
Элементов блочной памяти	1	268	0,4
Тактовых буферов	1	16	6
Умножителей-накопителей (DSP48a1s)	29	180	16

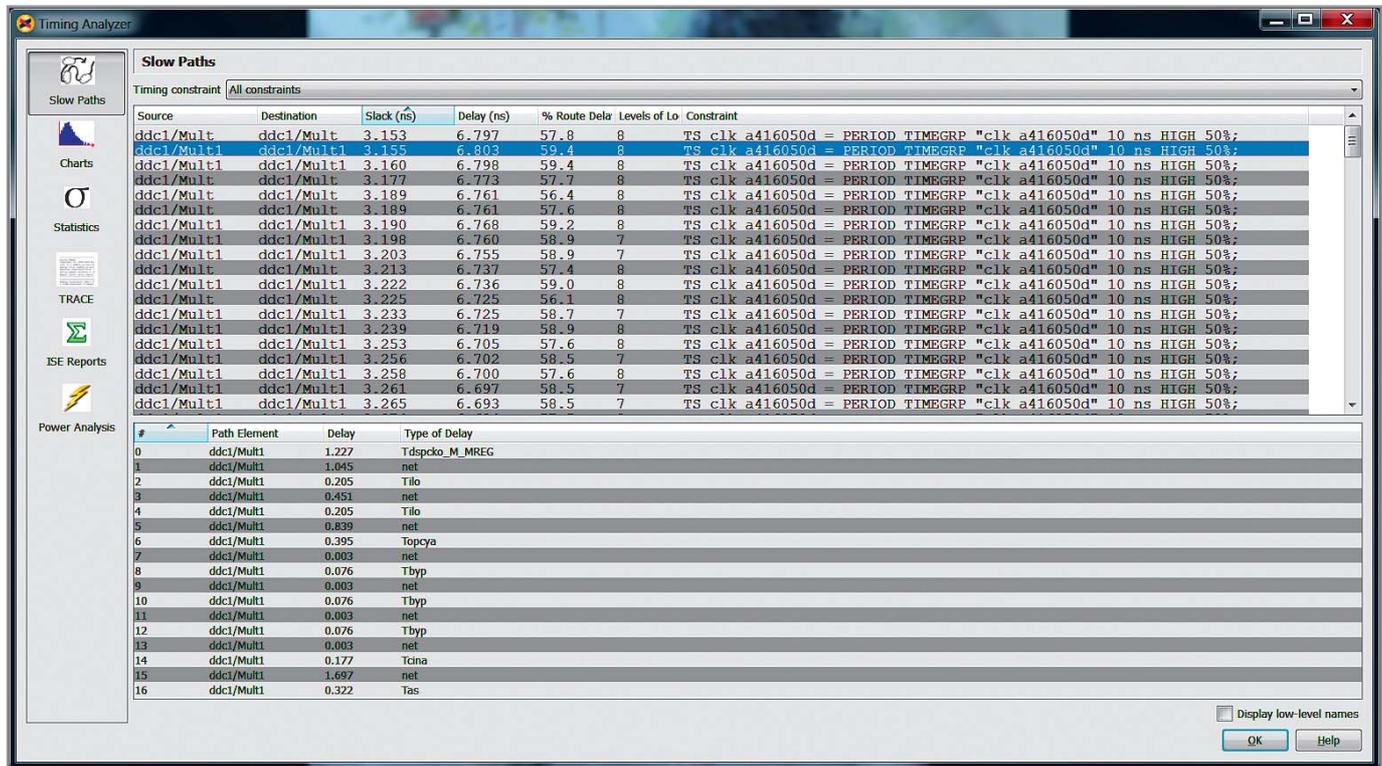


Рис. 11. Окно временного анализатора с задержками для выбранных элементов

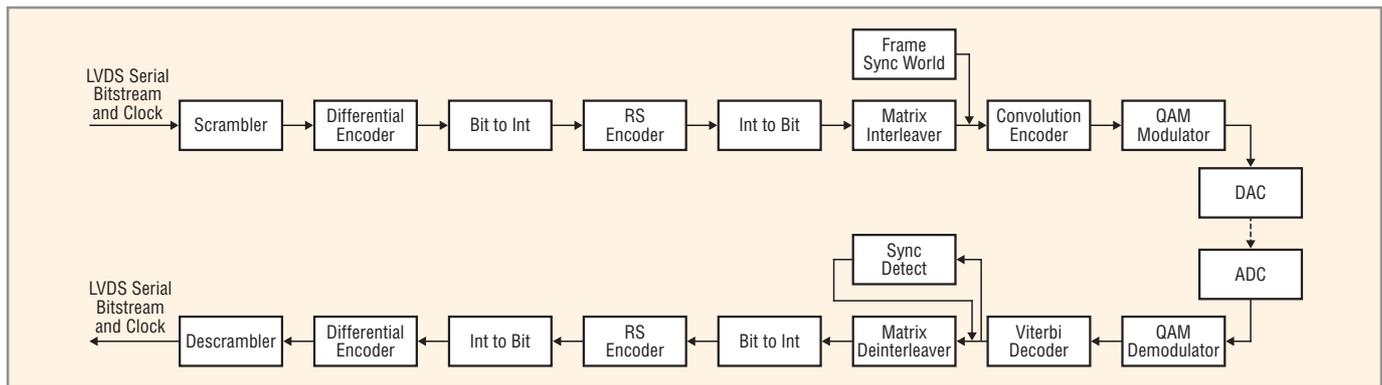


Рис. 12. Блок-схема спутникового приёмопередатчика

Итак, ПЛИС страссирована. Временных ошибок нет. Внизу приведён стандартный отчёт о ресурсах нашего проекта после компиляции ПЛИС (см. таблицу 2). Основную их часть «съел» двухканальный КИХ-фильтр длиной 226 отводов – 29 блоков умножителей-сумматоров DSP48A1s. Для многих проектов такое большое число умножителей может показаться критичным.

Попробуем перепроектировать наш фильтр на архитектуру с распределённой арифметикой и избавиться от умножителей в фильтре. Для этого выберем в блоке FIR Compiler архитектуру фильтра distributed arithmetic. Такая структура не поддерживает многоканальность, поэтому сделаем два фильтра (см. рис. 10). После трассировки оказалось, что число умножителей (DSP48A1s) уменьшилось почти на

порядок, но более чем в 20 раз возросло число логических ячеек LUT (см. таблицу 3). Таким способом можно эффективно управлять ресурсами проекта.

Можно провести анализ задержек (Timing and Power analysis) (см. рис. 11) и прямо в Simulink увидеть названия элементов из меню Slow Path, задержку в наносекундах, число уровней логики и т.д. Данный метод очень нагляден и позволяет быстро найти критическую цепь.

Таблица 3. Ресурсы проекта на ПЛИС XC6SLX150T после оптимизации числа умножителей

Название ресурса	Использовано	Всего	В % от ресурса
Количество триггеров	13 799	184 304	7,5
Число программируемых логических таблиц, использованных как логика (LUT)	12 799	92 152	14
Число программируемых логических таблиц, использованных как память (LUT)	1011	21 680	5
Элементов блочной памяти	1	268	0,4
Тактовых буферов	1	16	6
Умножителей-накопителей (DSP48a1s)	3	180	1,6

ПРИМЕР РЕАЛИЗАЦИИ БОЛЬШОЙ СИСТЕМЫ С ОЦЕНКОЙ ПРОИЗВОДИТЕЛЬНОСТИ

Компания BAE systems провела показательную разработку спутникового приёмопередатчика с использованием Xilinx System Generator [2]. Две группы разработчиков работали параллельно.

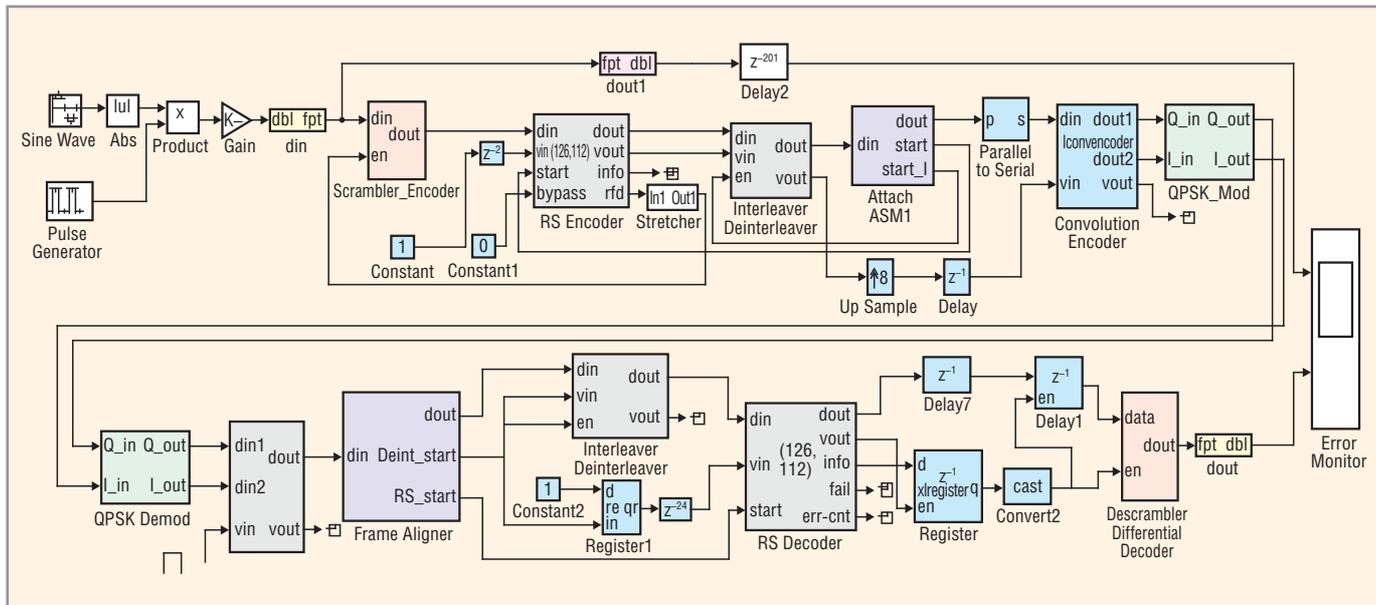


Рис. 13. Блок-схема проекта в Xilinx System Generator

Одна из групп действовала классическим методом под руководством специалиста по проектированию на языке VHDL, вторая, соответственно, использовала модельно-ориентированный метод на основе System Generator. Обе группы должны были разработать одинаковый проект спутникового приёмопередатчика (см. рис. 12). Обе группы должны были смоделировать алгоритм согласно спецификации, спроектировать и промоделировать модули, создать VHDL-описание проекта и осуществить реализацию в аппаратуре. Обе группы имели предварительно готовые и верифицированные блоки Рида-Соломана, Витерби и перемежителей.

Группе, работавшей на языке VHDL, потребовалось 782 ч на разработку.

Много времени ушло на выбор разрядности представления данных с фиксированной точкой и верификацию проекта с плавающей точкой.

Группе, использовавшей System Generator (см. рис. 13), потребовалось всего 45,5 ч. Этап кодирования в VHDL был осуществлён автоматически. Мощные средства визуализации системы помогли быстро осуществить моделирование и найти ошибки в проекте.

ПРИМЕРЫ В СОСТАВЕ ПАКЕТА

В состав пакета входит множество рабочих примеров, что упрощает практическое знакомство с системой:

- 16 QAM-демультиплексор;
- коррекция ошибок для стандарта DVB;

- петля Костаса;
- канал MIMO;
- преобразователь частоты для стандарта GSM;
- восьмиканальный полифазный фильтр;
- адаптивный эквалайзер LMS;
- фильтры для двумерной обработки видео;
- преобразование Фурье в реальном масштабе времени;
- примеры с плавающей запятой и вычислители CORDIC и т.д.

ЛИТЕРАТУРА

1. Xilinx System Generator for DSP User Guide. (www.xilinx.com).
2. Xilinx DSP Magazine (issue 2). (www.xilinx.com).

