

3973.20-08.2 + 3844.150.2-02-5=03

Палагин А. В. чл. - кор. НАН Украины

Опанасенко В. Н. канд. техн. наук

Сахарин В. Г. канд. техн. наук

Лисовый А. Н.

ИСПОЛЬЗОВАНИЕ УТИЛИТЫ SYSTEM GENERATOR ДЛЯ ВЕРИФИКАЦИИ ПРОЕКТОВ, РЕАЛИЗУЕМЫХ В ЭЛЕМЕНТНОМ БАЗИСЕ ПЛИС

Институт кибернетики им. В.М. Глушкова НАН Украины

Рассмотрены особенности использования утилиты System Generator, которая является связующим звеном между системами MATLAB/Simulink и Xilinx Foundation ISE и позволяет освободить разработчика от рутинной работы, связанной с детализацией проекта, произвести проверку функционирования проектируемого объекта на самой начальной стадии разработки и тем самым сократить сроки и повысить качество проектирования. Предложен подход к проектированию FIR-фильтра с отображением сигналов, формируемых для проверки функционирования проекта, и выполнена его реализация в элементном базисе ПЛИС.

Введение

Современное развитие элементной базы, в частности программируемых логических интегральных схем (ПЛИС), обеспечивает создание сложных устройств и систем в одном кристалле. Поэтому актуальной является задача верификации разрабатываемых проектов как с самого начала (описание алгоритмов функционирования устройства и разработка его логической модели), так и непосредственно на всех стадиях проектирования. При этом среда моделирования должна обеспечивать формирование набора входных сигналов, адекватных сигналам реальной рабочей среды проектируемого устройства, а также возможность визуализации сигналов, необходимых для его проверки и отладки. Другой актуальной задачей, вызванной сложностью разрабатываемых устройств, является задача описания проекта на достаточно высоком уровне представления (математическое или алгоритмическое представление) с последующей автоматической или полуавтоматической трансляцией этого представления в аппаратную модель в соответствующем элементном базисе.

Решение указанных задач может быть осуществлено с помощью широко распространенной системы MATLAB/Simulink и системы проектиро-

вания Foundation ISE фирмы Xilinx, связанных через утилиту System Generator.

Основным средством верификации проектов методом моделирования в современных системах проектирования ПЛИС типа FPGA фирмы Xilinx [1] является система ModelSim Xilinx Edition (MXE II) [2]. К наиболее важным свойствам ModelSim относятся:

- возможность поведенческого моделирования объектов, описанных на HDL-языке;
- высокая скорость моделирования;
- высокая (до 1 фемтосекунды) разрешающая способность;
- широкий выбор систем счисления и форматов представления данных (например, целых чисел в десятичной системе счисления со знаком, в том числе в аналоговом формате);
- возможность использования HDL-языка для подготовки файлов входных воздействий (Test Bench) и т.д.

Верификация может производиться на различных стадиях выполнения проекта, и поэтому различают четыре этапа или вида моделирования:

- поведенческое (Behavioral) функциональное моделирование, выполняемое для проверки правильности функционирования проекта путем интерпретации HDL-кода. Для такого вида часто исполь-

зуется термин *RTL (Register Transfer Level)* – уровень регистровых передач) моделирование;

- *функциональное моделирование* после выполнения синтеза (*Post-Translate*) на основе продуцированного системой синтеза списка цепей вентиляционного уровня;

- *моделирование после планировки (Post-Map)* с частичным учетом временных задержек (без учета задержек в цепях трассировки) на основе списка цепей вентиляционного уровня;

- *моделирование после размещения и трассировки (Post-Place & Route)* с полным учетом временных задержек на основе списка цепей вентиляционного уровня.

Для создания модели проектируемого устройства, переносимой в среду физического синтеза, широко используется система *MATLAB*, интегрированная с программным средством *Simulink*, созданным для моделирования динамических систем и устройств, задаваемых в виде совокупности блоков [3]. Принципы визуально-ориентированного программирования *Simulink* позволяют выполнять моделирование сложных устройств с достаточно высокой степенью достоверности и хорошим качеством представления результатов.

System Generator является средством расширения *Simulink* для моделирования логики, используемой в цифровой обработке сигналов, и трансляции *Simulink*-модели в аппаратную модель устройства, реализуемого в *Xilinx FPGA*. Модели создаются из *Xilinx*-блоков, поведение которых адекватно функционированию проектируемого устройства. В *Simulink* эти модели могут быть достаточно просто соединены с блоками, не встраиваемыми в *FPGA (non-Xilinx blocks)* для создания окружающей среды разрабатываемого устройства. Требуемые параметры *Xilinx*-блоков устанавливаются непосредственно из рабочей среды проектирования *MATLAB* аналогично параметрам *Simulink*-блоков. Поведение части *Simulink*-модели, не встраиваемой в *FPGA*, может быть преобразовано в *HDL*-описание входных воздействий (*Test*

Bench), используемое при моделировании проектируемого устройства. Это позволяет произвести сравнение функционирования поведенческой модели, созданной в среде *Simulink*, с аппаратной моделью устройства. Создание *Simulink*-модели с ее последующей автоматической трансляцией в аппаратную модель облегчает процесс разработки для проектировщиков, не в полной мере владеющих приемами работы с проектными средствами *Xilinx*.

Проектирование устройств в среде *MATLAB*

Схема процесса проектирования с использованием *System Generator* [4] представлена на рис. 1.

Разработка проекта начинается с математического описания операций, содержащихся в алгоритме аппаратной реализации. При проектировании с применением *System Generator* предусматривается три типа представления арифметических данных, обычно используемых в *DSP*: с плавающей точкой двойной точности, фиксированной точкой со знаком и без знака. Данные с плавающей точкой не могут быть автоматически преобразованы в аппаратную модель и операции над ними выполняются только в процессе моделирования.

Для согласования *Simulink*-блоков с *Xilinx*-блоками в *System Generator* имеются шлюзы (*Gateway*-блоки), выполняющие преобразование из одного формата представления данных в другой. Большинство *Xilinx*-блоков являются полиморфными, т.е. обладают возможностью реконфигурации, заключающейся в изменении их типов выходов в зависимости от типов входов. Путем выбора соответствующих параметров можно получить на выходе требуемую точность представления данных с учетом эффектов квантования и переполнения разрядной сетки. Естественно, что получение более высокой точности результата операции связано с увеличением аппаратных затрат при реализации проекта в кристалле.

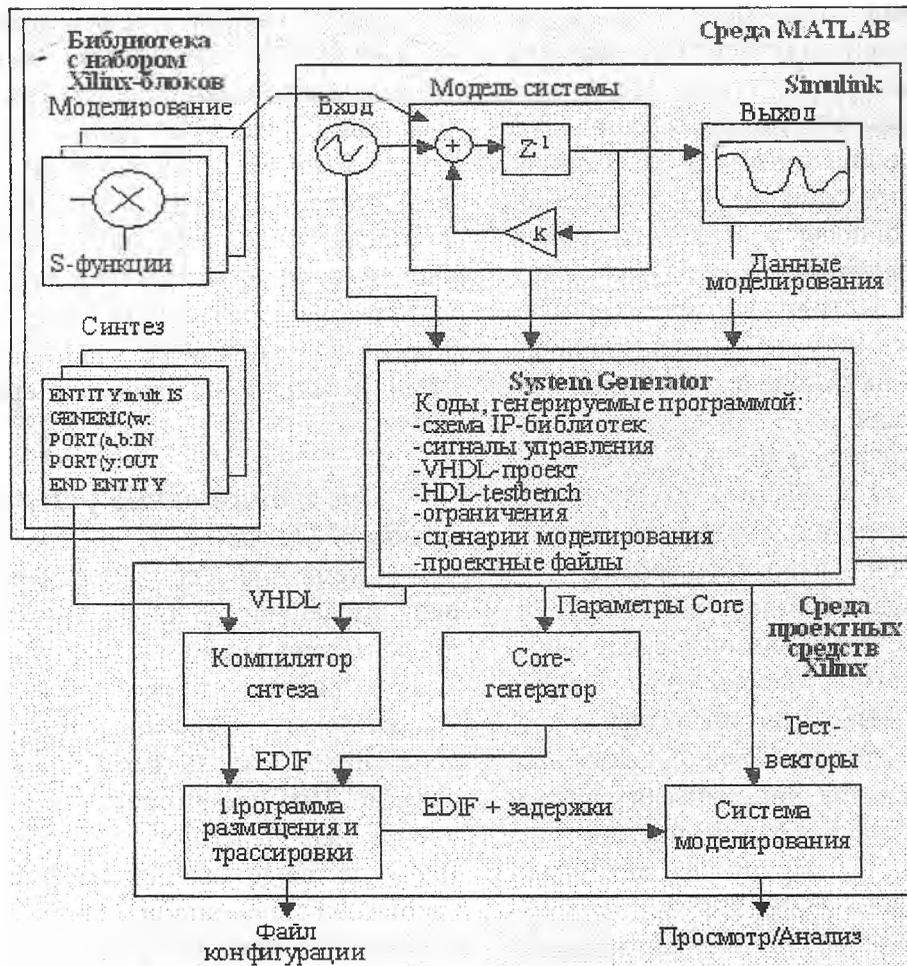


Рис. 1. Схема процесса проектирования с использованием *System Generator*

Реализация проекта в среде *System Generator* предполагает единственную тактирующую последовательность (*clock*), которая является задающей синхропоследовательностью для всех элементов схемы, содержащих триггеры и/или память. Для проектов, имеющих различные частоты синхронизации, используется сигнал разрешения (*clock enable*), который активирует синхропоследовательности на требуемых интервалах времени. Каждый период выборки (шаг дискретизации) в *Simulink*-модели может быть связан с соответствующим сигналом разрешения.

Любой сигнал в среде *System Generator* является дискретным, шаг дискретизации которого наследуется в соответствии с правилами функционирования *Simulink* или устанавливается отдельно для блока посредством диалогового меню. Обычно каждый блок определяет частоту выборки на входе и порождает соответствующую частоту на выходе. В наборе *Xilinx*-блоков предусмотрены также

способы увеличения и уменьшения частоты выборки с помощью блоков *Up Sample* и *Down Sample*.

Набор *Xilinx*-блоков содержит 9 групп (библиотек), сформированных по функциональному признаку. Для удобства, наиболее часто используемые блоки (например, *Gateway*-блоки) располагаются в нескольких, разных библиотеках. Полный набор блоков содержит *Index*-библиотека, в которой все блоки расположены в алфавитном порядке. Кроме этой библиотеки имеются следующие: базовые элементы (стандартные встраиваемые блоки цифровой логики), элементы коммуникации (блоки цифровых систем связи), логика управления (блоки схем управления и конечные автоматы), типы данных (блоки для преобразования типов данных), математика (блоки реализации математических функций), *DSP* (блоки систем цифровой обработки сигналов), память (блоки реализации памяти), сервис (функциональные утилиты).

Каждая разрабатываемая схема должна содержать блок, называемый *System Generator*, предназначенный для управления системой параметров и вызова генератора кодов, формирующего проектные файлы. Библиотеки содержат такие блоки как аккумулятор, адресуемый сдвиговый регистр, мультиплексор, сумматор/вычитатель, умножитель, счетчик, цифровой синтезатор с прямым синтезом частот (*DDS*), элемент задержки, однопортовая и двухпортовая память (*RAM*), память *FIFO*, постоянная память (*ROM*), фильтр с конечной импульсной характеристикой (*FIR*), формирователь синусоидального/косинусоидального сигнала, декодер *Viterbi*, микроконтроллер *PicoBlaze* и др. Для разработки каких-либо оригинальных устройств или систем на языке *VHDL* может быть использован блок «черный ящик» (*Black Box*), для которого подготавливается файл с текстом описания функционирования блока на языке *VHDL*.

Таким образом, алгоритм или математическое описание выполняются в виде схемы (модели), компонуемой в среде *MATLAB/Simulink* из *Simulink*-блоков и *Xilinx*-блоков. После моделирования проекта в этой среде инициализируется генератор кодов, формирующий в соответствующих каталогах полный проект для системы *Xilinx Foundation ISE*. При инициализации блока *System Generator* указываются: серия и тип кристалла, в котором предполагается разместить проект (*Virtex-II Pro*, *Virtex-II*, *Virtex-E*, *Virtex*, *Spartan-IIe*, *Spartan-II*, *Spartan-3*), тип инструментального средства синтеза (*Leonardo Spectrum*, *Synplicity Synplify Pro*, *Synplify*, *Xilinx XST*), корневой каталог, в котором должны быть размещены проектные файлы, временные соотношения (системный период *Simulink*, представляющий собой наибольший общий делитель для периодов выборки в модели, и период синхронности в кристалле), необходимость формирования теста на языке *VHDL* для моделирования в системе *ModelSim*, возможность использования ядер, формируемых системой *Xilinx LogiCORE*. Далее завершение проекта осуществляется в системе *Xilinx Foundation ISE*, использующей подготовленные проектные файлы. При необходимости производится

верификация проекта в системе *ModelSim* для выбранных этапов моделирования.

Реализация Xilinx-блоков в кристаллах ПЛИС

Рассмотрим схему (рис. 2) для формирования сигнала синусоидальной формы.

Счетчик (*Xilinx*-блок *Counter*) генерирует 8-разрядный вектор кодов, поступающий на входы *Xilinx*-блока *SineCosine*. Для визуализации полученного результата к выходу блока *SineCosine* через шлюз *Gateway Out* подключены *Simulink*-блоки *Scope* (осциллограф) и *XY Graph* (графопостроитель). Формирование временной развертки графопостроителя выполняется с помощью *Simulink*-блока *Digital Clock*.

Реализация этой схемы (*Xilinx*-блоков) в кристалле *Spartan-II XC2S50TQ144* требует 24 триггера и 75 4-входовых *LUTs* или 31 слайс из 768 имеющихся слайсов, что составляет 4% ресурсов кристалла. Рассмотрим процесс (рис. 3) проектирования *FIR*-фильтра с отображением сигналов, формируемых для проверки функционирования проекта.

Визуализация сигналов производится блоками *XY Graph* с выходов блоков: генератора сигналов синусоидальной формы (*Sine Wave*), генератора сигналов с линейной частотной модуляцией (*Chirp Signal*), сумматора сигналов этих генераторов, фильтра (*FIR*). Форма полученного сигнала с выхода *FIR*-блока позволяет оценить работоспособность фильтра.

Проектирование фильтра производится с помощью инструментального средства *Filter Design and Analysis Tool (FDATool)* системы *MATLAB/Simulink*, размещаемого в виде *Xilinx*-блока на схеме. Параметры фильтра, относящиеся собственно к выбору типа фильтра (низкие частоты, высокие частоты, полоса частот пропускания, полоса частот затухания), его структуры, порядка, частот дискретизации и отсечки, выбираются путем инициализации блока *FDATool*. Параметры, относящиеся к выбору разрядности коэффициентов и конструктивным особенностям реализации фильтра в кристалле, задаются при инициализации блока *FIR*. На рис. 4. Приведены внешний вид окна *FDATool* и амплитудно-частотная характеристика проектируемого фильтра, полученная при работе с указанным средством.

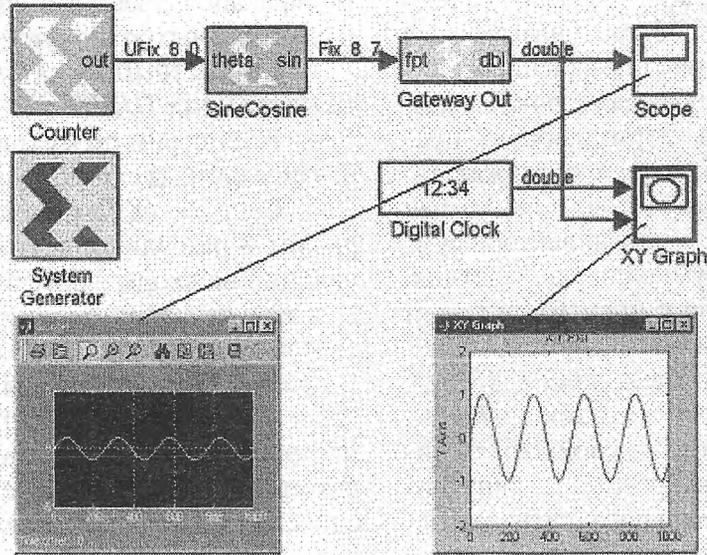


Рис. 2. Реализация схемы в среде *Simulink/Xilinx* с отображением результата моделирования

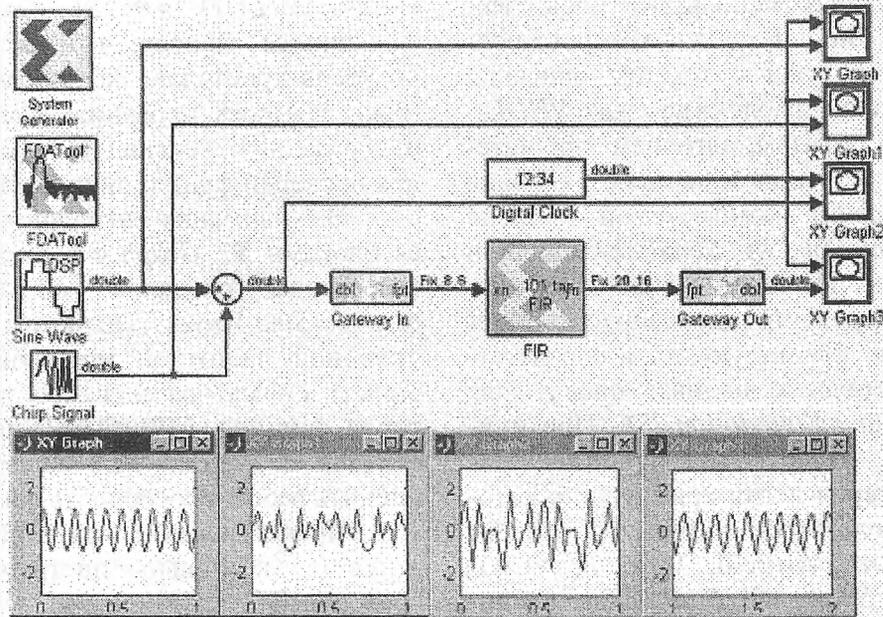


Рис. 3. Реализация схемы *FIR*-фильтра в среде *Simulink/Xilinx* с отображением результатов моделирования

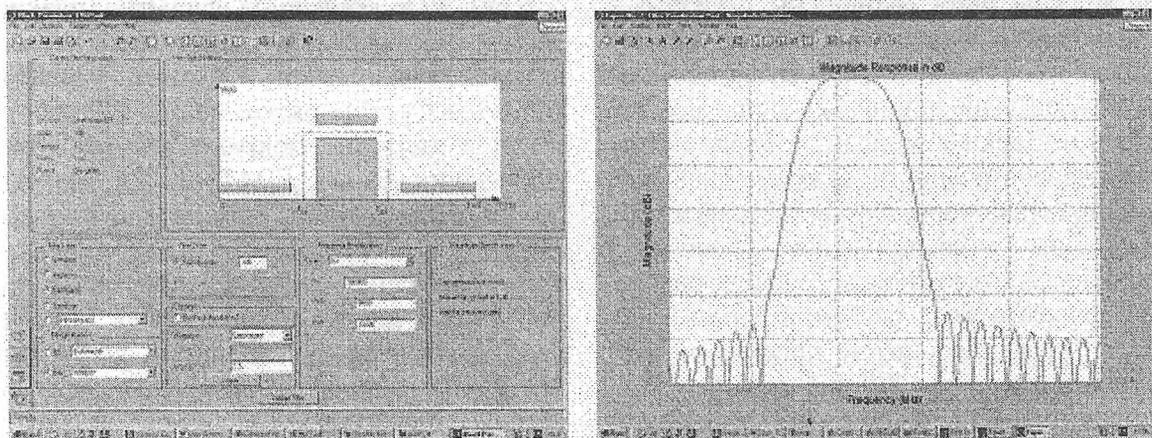


Рис. 4. Внешний вид окна *FDATool* и АЧХ проектируемого фильтра

Выходным продуктом *FDA Tool* является файл с рассчитанными коэффициентами, используемыми далее для синтеза. Количество коэффициентов определяется числом звеньев фильтра, т.е. N -звенному фильтру соответствует набор коэффициентов $h(0), h(1), \dots, h(n-1)$, каждый из которых представлен числом с фиксированной точкой. Поток данных $x(0), x(1), \dots$, поступающих на вход фильтра, продуцирует на его выходе данные, рассчитываемые по формуле:

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i).$$

В рассмотренном примере выбран полосовой гауссовский фильтр с порядком, равным 100, частотой дискретизации

$F_s=100$ КГц, частотами отсечки $F_{c1}=8400$ Гц и $F_{c2}=13200$ Гц. На вход фильтра подавалась сумма синусоидального сигнала с частотой 10 КГц и синусоидального сигнала с линейной частотной модуляцией от 1 МГц до 20 КГц за 0,1мс.

На рис. 5 приведено отображение результата моделирования фильтра в системе *ModelSim*.

Реализация *Xilinx*-блока этого фильтра с параллельной структурной организацией в кристалле *Spartan-II* *XC2S300EPQ208* заняла 6458 слайсов типа *Flip-Flop*, 5317 слайсов типа 4-входовых *LUTs* при синтезе и 2228 слайсов из 3072 имеющихся слайсов при размещении и трассировке проекта, что составляет 72% ресурсов кристалла.

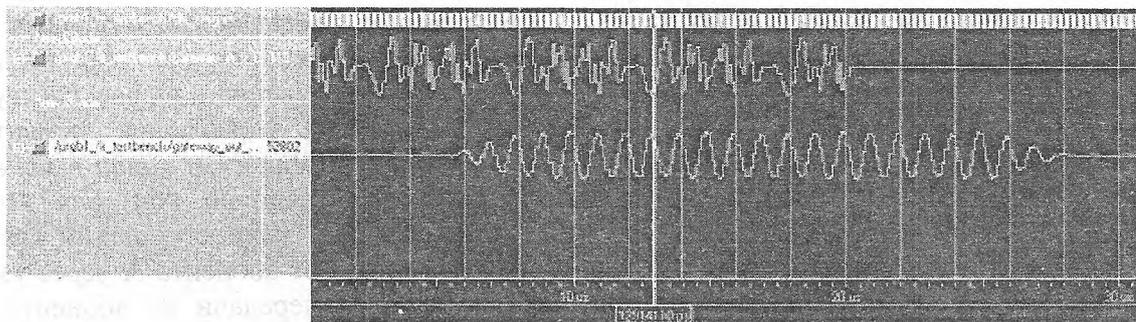


Рис. 5. Отображение результата моделирования фильтра в системе *Model Sim*

Выводы

ПЛИС находят широкое применение при разработке современных устройств и систем управления и вычислительной техники. Например, в цифровой обработке сигналов, по сравнению с сигнальными процессорами, ПЛИС имеют такие преимущества, как возможность организации параллельной обработки данных, масштабирование полосы пропускания, расширяемость устройства. Увеличение логической емкости, усовершенствование архитектурных возможностей и потребительских свойств ПЛИС требует соответствующего усовершенствования инструментальных средств для разработки проектов на их основе.

Использование утилиты *System Generator*, которая является связующим звеном между системами *MATLAB/Simulink* и *Xilinx Foundation ISE*, позволяет освободить

разработчика от рутинной работы, связанной с детализацией проекта, произвести проверку функционирования проектируемого объекта на самой начальной стадии разработки и тем самым сократить сроки и повысить качество проектирования.

Список литературы

1. Available at <http://www.xilinx.com/literature/index.htm>.
2. *ModelSim* / available at <http://www.asicdesign.ru/Presentations/modelsim/modelsim1.html>.
3. Дьяконов В.П. *MATLAB 6/6.1/6.5 + Simulink 4/5* в математике и моделировании. Полное руководство пользователя. - М.: СОЛОН-Пресс. - 2003. - 576 с.
4. *Xilinx System Generator for DSP. Version 3.1. User Guide Xilinx Block set. Reference Guide* / available at <http://www.xilinx.com>.