УДК 004.414.2

АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ УСТРОЙСТВ ГЕНЕРАЦИИ КОМБИНАТОРНЫХ ПЕРЕСТАНОВОК

Зольников Владимир Константинович
д.т.н., профессор

Соболев Сергей Сергеевич
аспирант
Воронежская государственная лесотехническая академия, Воронеж, Россия

В статье представлены модели, алгоритмы, методы автоматизации проектирования специализированных устройств генерации полных комбинаторных перестановок символьной строки переменной разрядности. Предложена методика проектирования таких устройств, обеспечивающая высокое качество проекта и наибольшую автоматизацию процесса проектирования на системном уровне и уровне регистровых передач (RTL-уровне). В конце статьи произведена оценка эффективности предложенных решений

Ключевые слова: САПР, КОМБИНАТОРНЫЕ ПЕРЕСТАНОВКИ, СИСТЕМНЫЙ УРОВЕНЬ ПРОЕКТИРОВАНИЯ, RTL-УРОВЕНЬ ПРОЕКТИРОВАНИЯ, SYSTEMC, HDL-ОПИСАНИЕ, VHDL

UDC 004.414.2

DESIGN AUTOMATION OF PERMUTATIONS GENERATORS

Zolnikov Vladimir Konstantinovich
Dr.Sci.Tech., professor

Sobolev Sergey Setrgeevich
postgraduate student
Voronezh State Academy of Forestry and Technologies, Voronezh, Russia

The models, algorithms, methods of design automation of specialized devices of permutations generation within the variable width string are presented in this article. The methodology of design of such devices, which provides high-quality project and maximal automation of design process at system level and register-transfer level (RTL) is offered. At the end of the article, the efficiency of the offered solutions is estimated

Keywords: EDA, COMBINATORIAL PERMUTATIONS, SYSTEM LEVEL, REGISTER-TRANSFER LEVEL, SYSTEMC, HARDWARE DESCRIPTION, VHDL

Специализированные устройства генерации полных комбинаторных перестановок символьной строки составляют специфический класс сверхбольших интегральных схем (СБИС). Такие устройства представляют интерес в системах аппаратной поддержки технологий систем автоматизированного проектирования радиоэлектронной аппаратуры [1]; защиты информации, [2, 3]; аппаратного обеспечения динамического преобразования форматов [4]. Характерными их чертами являются переменная разрядность символьной строки и наличие ограничений на требуемое множество генерируемых комбинаторных перестановок в некоторых приложениях.

Применение стандартных средств проектирования при разработке рассматриваемых специализированных СБИС связано с рядом проблем. Современная методология проектирования предполагает разбиение маршрута проектирования на несколько этапов проектных процедур. Первыми

двумя этапами являются проектирование на системном уровне и на уровне регистровых передач (RTL-уровень). На верхнем структурном уровне, называемом системным, разрабатывается и верифицируется обобщенная модель блока для проверки работоспособности взятого за основу математического алгоритма генерации перестановок [5]. В качестве языка описания чаще всего используется язык SystemC [6]. На RTL-уровне разрабатывается функциональное описание блока на уровне регистровых передач с использованием одного из языков описания аппаратуры (HDL-языков). Такое описание осуществляется чаще всего вручную, путем графического или текстового ввода, что, во-первых, увеличивает трудоемкость и требует недопустимо больших затрат времени, а во-вторых, влечет за собой разрыв между алгоритмическим описанием на SystemC и RTL-описанием. Аналогичный разрыв наблюдается и при разработке тестов для верификации блоков на системном и RTL уровнях. Обозначенные проблемы, вкупе с многообразием входных параметров генерации комбинаторных перестановок и увеличением сложности схемы блока генерации с ростом длины символьной строки, повышают риск субъективных ошибок и, таким образом, увеличивают влияние «человеческого фактора» на качество всего проекта. В результате, использование традиционной методологии в процессе проектирования рассматриваемых устройств представляется затруднительным. В этой связи, актуальной задачей представляется разработка средств автоматизации проектирования специализированных устройств генерации комбинаторных перестановок элементов символьной строки. В настоящей статье представлены методика, модели, методы и алгоритмы для автоматизации проектирования рассматриваемых целевых устройств, обеспечивающие наибольшую автоматизацию процесса проектирования на системном и RTL-уровнях, а также высокое качество всего проекта. Кроме того, приводятся результаты использования предлагаемых решений.

Методика проектирования устройств генерации комбинаторных перестановок. Важнейшим принципом, на котором основывается методика проектирования современных СБИС, является использование в проекте уже готовых функционально законченных блоков (СФ-блоков, или IP-блоков). Специфика устройств генерации полных комбинаторных перестановок элементов символьной строки с отсевом недопустимых перестановок связана с их следующими особенностями: длина символьной строки, а также разрядность одного элемента символьной строки, для которой производится генерация комбинаторных перестановок, является переменной величиной; максимально допустимое количество неподвижных точек и/или инволюций является переменной величиной.

Учитывая данную специфику, в рамках проектирования на системном уровне предложено: в качества языка описания поведенческой модели использовать SystemC – язык проектирования и верификации моделей системного уровня; генерировать данную поведенческую модель с помощью программного модуля, реализованного на высокоуровневом языке C++, на основе следующих входных параметров: разрядность одного элемента символьной строки, для которой производится генерация комбинаторных перестановок; длина данной символьной строки, или мощность множества, для которого формируются перестановки; количество неподвижных точек и инволюций в каждой из сгенерированных перестановок, достаточное для ее отсеивания; генерировать тесты для верификации поведенческой модели также с помощью с помощью программного модуля на C++ с тем же множеством входных параметров.

Полученные на системном уровне спецификации являются исходной точкой для следующего этапа: проектирование цифровых, цифро-аналоговых и аналоговых блоков. Отметим, что проектирование каждого из этих блоков может осуществляться параллельно. Остановимся более подробно на последующем маршруте проектирования цифрового блока ге-

нерации перестановок. Этот уровень часто называют логическим, или вентильным. На этапе разработки функционального описания блока на уровне регистровых передач (RTL) создается синтезируемая RTL-модель блока на одном из языков описания аппаратуры. В рамках проектирования рассматриваемых устройств генерации перестановок, в качестве такого языка предложен язык VHDL [7, 8]. Данный этап выполняется в автоматическом режиме, предполагающем генерацию VHDL-кода с помощью программного модуля на языке C++. Учитывая специфику блока генерации полных комбинаторных перестановок элементов символьной строки, модуль принимает на вход то же множество входных параметров, что и для автоматизированного формирования SystemC модели на системном уровне.

Следующий этап включает в себя моделирование и верификацию RTL-модели. Верификация модели предполагает прохождение предварительно сформированного набора тестов, специфичных для блока генерации перестановок. Тесты для верификации на RTL-уровне генерируются также с помощью программного модуля на языке C++ с тем же множеством входных параметров, что и для модели на системной уровне. Данный метод позволяет произвести откат к системному уровню в случае неудачи верификации. Примером может послужить ситуация, когда какие-либо параметры, заложенные в блок на системном уровне, не реализуются, и тогда можно изменить саму системную модель, внеся необходимые правки в программный модуль генерации модели. Дальнейший маршрут проектирования включает в себя логический синтез, логическое моделирование и верификацию, физический синтез и верификацию топологии.

Метод генерации функциональных блоков. Для обеспечения гибкости системной и RTL моделей генератора перестановок используется множество входных параметров, позволяющих по требованию модифицировать основные характеристики устройства. На основе этих параметров генерируются поведенческая модель на SystemC и RTL-модель на VHDL,

отвечающие всем входным аргументам. Множество параметров, позволяющих кастомизировать указанные модели, включает в себя: d – разрядность одного элемента символьной строки, для которой производится генерация комбинаторных перестановок (в битах); n – длина данной символьной строки, или мощность множества, для которого формируются перестановки; $N_{\max}$ – максимальное количество неподвижных точек в каждой из сгенерированных перестановок, при котором перестановка не отсеивается; $R_{\max}$ – максимальное количество инволюций в каждой из сгенерированных перестановок, при котором перестановка не отсеивается;

Для достижения большей адаптивности моделей, их генерация происходит в несколько этапов. Первый этап – это формирование набора процедур, каждая из которых может принимать в качестве входных аргументов один или несколько вышеприведенных параметров, а также ряд других опциональных параметров. Каждая процедура позволяет настраивать отдельные части моделей и регулировать взаимодействия между ними.

Второй этап заключается в синтаксическом разборе списка процедур и генерации кода конечных моделей на SystemC и VDHL. Эти действия осуществляет программный модуль на языке высокого уровня C++.

Общая схема двухступенчатой генерации функциональных блоков изображена на рис. 1. Верификация системной модели и RTL-модели производится с помощью специализированного тестового окружения под управлением специализированной подсистемы верификации.

Алгоритм генерации комбинаторных перестановок с отсевом не допустимых перестановок. Для генерации комбинаторных перестановок элементов символьной строки длины n с возможностью отсева слабых перестановок предложен алгоритм на основе циклического сдвига, представляющий собой комбинацию следующих последовательно применяемых элементарных операций: добавление нового символа в строку (увеличение

полезной длины строки) и циклический сдвиг строки [9]. Условная схема алгоритма представлена на рис. 2.

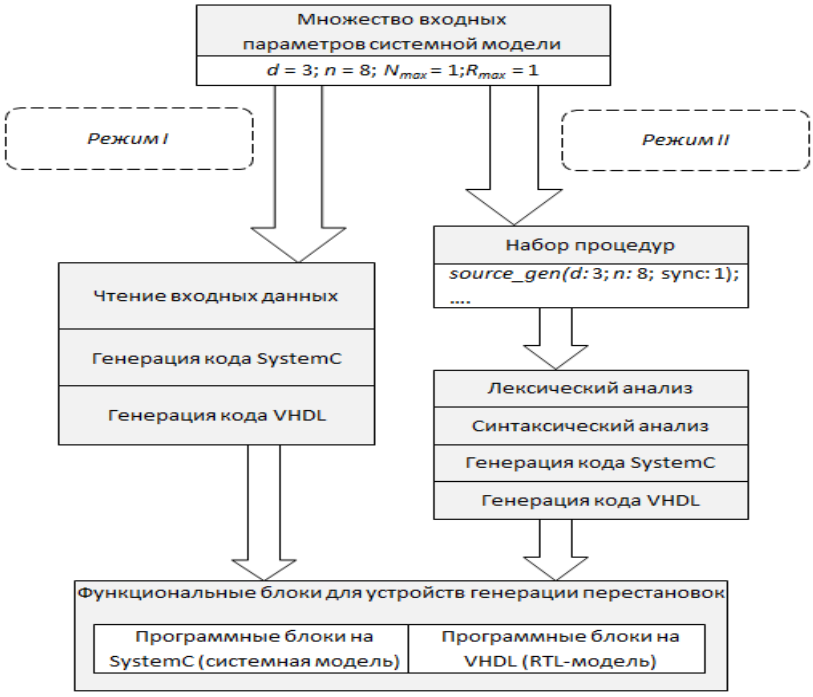

Рис. 1. Схема двухступенчатой генерации функциональных блоков генерации перестановок

Рис. 2. Схема алгоритма генерации перестановок для $n = 3$.

Моделирование генерации перестановок на системном и RTL уровнях. Общая структура подсистемы генерации полных комбинаторных перестановок представлена на рис. 3. Она включает в себя несколько блоков, каждый из которых имеет свою функциональность.

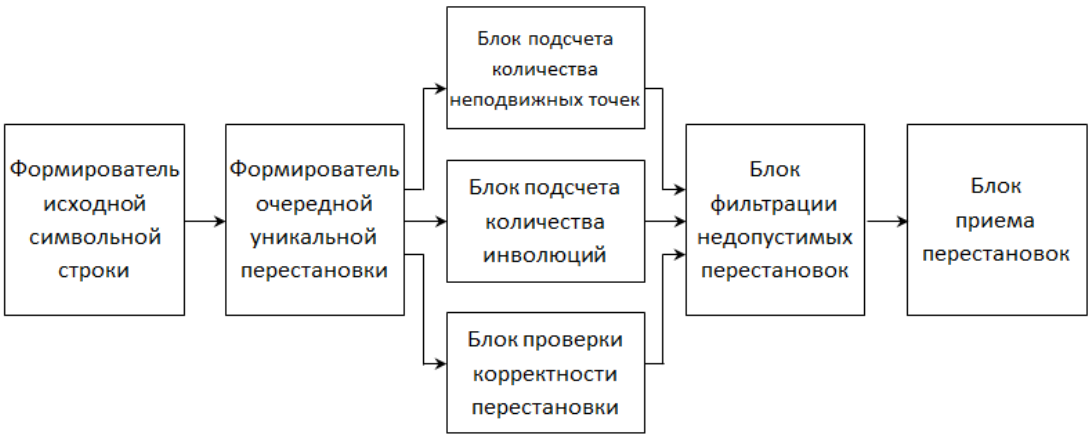

Рис. 3. Общая структура блока генерации полных комбинаторных перестановок с отсевом недопустимых перестановок

Каждому блоку на рис. 3 соответствует отдельный модуль в системной модели SystemC.

Модуль формирования очередной уникальной перестановки включает в себя $n-1$ модулей циклического сдвига и $n-1$ вспомогательных модулей управления. Модули циклического сдвига соединены последовательно и имеют уровень от 2 до n в соответствии с увеличением полезной длины символьной строки согласно алгоритму на основе циклического сдвига. Каждый модуль циклического сдвига обладает портами управления *L* и *S*. При высоком логическом уровне сигнала на входе *L* значения элементов символьной строки считываются с входных портов и тут же передаются на выходные порты (либо на следующий уровень $m+1$, либо в качестве результирующей перестановки), без циклического сдвига. При высоком логическом уровне сигнала на входе *S*, происходит циклический сдвиг строки, хранящейся во внутреннем массиве модуля, и её передача на выходные порты. Значения сигналов на портах *L* и *S* вычисляются соглас-

но значениям функций $L = L(c,m,n)$ и $S = S(c,m,n)$, где m – уровень модуля, а c – внутренний счетчик вспомогательного модуля управления, увеличивающийся с каждым тактовым импульсом.

$$L(c,m,n) = \left[c \bmod \left(\prod_{i=m}^{n} i\right) = 0\right],$$

$$S(c,m,n) = \begin{cases} \overline{L(c,m,n)}, & m = n \\ \overline{L(c,m,n)} \wedge \left[\left(c \bmod \left(\prod_{i=m}^{n} i\right)\right) \bmod \left(\prod_{i=m+1}^{n} i\right) = 0\right], & m < n \end{cases}.$$ (

Разработка RTL-описания блока генерации полных комбинаторных перестановок с отсевом недопустимых перестановок осуществлялась с применением языка описания аппаратуры VHDL [7, 8]. Базовым элементом любого проекта на VHDL является объект. Блок генерации перестановок представляет собой совокупность моделируемых объектов VHDL, каждый из которых соответствует отдельному блоку на общей схеме, изображенной на рис. 3. Формирователь исходной символьной строки представляет собой $(n \cdot d)$-разрядный регистр с параллельным вводом и выводом. Регистр хранит в себе элементы исходного упорядоченного множества X_0, на основе которого осуществляется генерация перестановок.

Формирователь очередной уникальной перестановки представляет собой набор регистров циклического сдвига с параллельным вводом и выводом, работа которых координируется вспомогательными объектами управления. Всего объект формирования уникальной перестановки содержит $d \cdot (n-1)$ регистров ($n-1$ групп по d регистров) сдвига и $n-1$ вспомогательных объектов управления. Ширина регистров в каждой группе ступенчато увеличивается от до n бит с шагом 1 в соответствии с увеличением полезной длины символьной строки согласно алгоритму на основе циклического сдвига. Каждый вспомогательный объект управления

координирует работу своей группы из d регистров с помощью сигналов управления, поступающих на входные управляющие порты регистров.

Для управления элементами перестановки, всякий регистр, помимо информационных портов, также обладает и двумя входными портами управления L и S. По переднему фронту сигнала на порте L регистр зачитывает соответствующие биты элементов перестановки с входных информационных портов. По переднему фронту сигнала на порте S в регистре осуществляется сдвиг на 1 бит.

Результаты. Для автоматизации проектирования семейства генераторов полных комбинаторных перестановок разработан комплекс проблемно-ориентированного программного обеспечения, внедряемый в общий маршрут проектирования данных устройств. Программное обеспечение включает в себя набор инструментальных средств для решения задачи автоматизации проектирования целевых устройств генерации перестановок на системном (поведенческом) уровне и уровне регистровых передач (RTL-уровне). Для оценки эффективности разработанных средств, определена сложность логической схемы каждого из семейства аппаратных генераторов перестановок путем подсчета общего количества логических вентилей и строк VHDL-кода для соответствующих устройств. В соответствии с полученными результатами произведена оценка временных затрат проектирования с учетом средней скорости разработки 100 вентилей в день (эта оценка является стандартной и неизменной на протяжении последних пяти лет) [10]. Результаты представлены в таблице 1.

Таким образом, автоматизация процессов аппаратного описания и верификации на системном и RTL уровнях позволяет существенно сократить цикл проектирования рассматриваемых устройств за счет исключения процесса ручного описания на одном из HDL-языков. Как отражено в таблице 1, этот процесс обладает значительной трудоемкостью, возрастающей в квадратичной зависимости с увеличением длины строки n.

Таблица 1 – Сложность логических схем устройств генерации комбинаторных перестановок и оценка временных затрат проектирования

	Длина символьной строки *n*		
	***n* = 16**	***n* = 24**	***n* = 32**
Количество логических вентилей	3200	7400	13300
Количество строк VHDL-кода	1120	2180	3780
Временные затраты, в человеко-днях (стандартная методика)	32	74	133
Временные затраты, в человеко-днях (предложенная методика)	1	1	1

Литература

1. Курейчик, В. М. Комбинаторные аппаратные модели и алгоритмы в САПР [Текст] / В. М. Курейчик, В. М. Глушань, Л. И. Щербаков; М.: Радио и связь. – 1990.

2. U. S. Pat. No. 5,734,721 Bl H04L9/00 Anti-spoof without error extension (ANSWER)/Clark James Monroe [Текст]. – March 31, 1998.

3. Ritter, T. Transposition Cipher with Pseudo-Random Shuffling: The Dynamic Transposition Combiner [Текст] / T.Ritter // Cryptologia. – 1991. V. 15 (1). – P. 1-17.

4. Молодченко, Ж. А. Динамическое форматирование представлений объектов реляционных СУБД на основе кластерных транспозиций [Текст] / Ж. А. Молодченко, Л. С. Сотов, В. Н. Харин // Естественные и технические науки. – М.: Изд-во компании "Спутник +". – 2007, № 6 (32). – С. 224-226.

5. Немудров, В. Системы-на-кристалле. Проектирование. Проектирование и развитие [Текст] / В. Немудров., Г. Мартин – Москва: Техносфера, 2004. – 216 с.

6. IEEE Std 1666-2005 SystemC Language Reference Manual [Текст] / The Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue, New York, NY 10016-5997, USA. – 2008.

7. IEEE Std 1076-2008 VHDL Language Reference Manual [Текст] / The Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue, New York, NY 10016-5997, USA. – 2009.

8. Бибило, П.Н. Системы моделирования интегральных схем на основе языка VHDL. StateCAD, ModelSim, LeonardoSpectrum [Текст] / П.Н. Бибило. – М.: СОЛОН Пресс, 2005. – 384 с.

9. Соболев, С. С. Алгоритм локализации полных комбинаторных перестановок с применением циклического сдвига [Текст] / С. С. Соболев, В. Н. Харин, Л. С. Сотов; Федеральное агентство по образованию, Государственное образовательное учреждение высшего профессионального образования, Воронежская государственная лесотехническая академия – Воронеж, 2010. – 6 с. : ил. – Библиогр.: с. 6. – Деп. в ВИНИТИ РАН 02.08.2010 №480-В2010.

10. Keating, M. Reuse Methodology Manual, Third Edition [Текст] / Michael Keating, Pierre Bricaud. – Kluwer Academic Publishers. – Boston/Dordrech/London. – 2002.