

ПРОГРЕСИВНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

ПРОГРЕССИВНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

PROGRESSIV INFORMATICS TECHNOLOGIES

УДК 004.2

Гриценко А. А.¹, Зеленева И. Я.², Сероштан С. Ю.³, Татолов Е. Р.⁴

^{1, 3, 4}Аспирант Донецкого национального технического университета
²Канд. техн. наук, доцент Донецкого национального технического университета

МЕТОДИКА ИССЛЕДОВАНИЯ СПОСОБОВ ОПТИМИЗАЦИИ УПРАВЛЯЮЩИХ АВТОМАТОВ В БАЗИСЕ ПЛИС FPGA

В статье предложена методика, позволяющая исследовать известные способы оптимизации управляющих автоматов для базиса современных ПЛИС FPGA. Показана структура комплекса программных средств, необходимых для реализации методики, а также представлены результаты ее апробации.

Ключевые слова: ПЛИС, управляющий автомат, FPGA, синтез, верификация, САПР.

ВВЕДЕНИЕ

В настоящее время известно большое количество методов оптимизации логических схем управляющих автоматов (УА), которые, в частности, рассмотрены в работах [1, 2]. Отметим, что управляющие автоматы имеют широкое применение и, как следствие, реализуются в различных элементных базисах.

На данный момент одним из наиболее распространенных элементных базисов являются программируемые логические интегральные схемы (ПЛИС), которые относятся к классу FPGA [3]. Использование данного базиса было актуально и раньше [3], на что указывают соответствующие разделы в [1] и [2]. Однако сейчас значимость ПЛИС FPGA возросла, что подтверждается, в частности, более детальным их рассмотрением в [4].

Исходя из этого, можно сформулировать задачу, которая заключается в адаптации ранее разработанных и апробированных методов оптимизации схем управляющих автоматов к базису современных ПЛИС FPGA. Проблема адаптации является достаточно сложной, если учитывать, что в каждом конкретном случае она имеет специфичное решение. Тем не менее, при подготовке к решению данной проблемы можно выделить процесс, который необходимо выполнять независимо от того, какой именно метод адаптируется в конкретном случае. Речь идет об исследовании результатов применения кон-

кретного метода оптимизации для базиса ПЛИС FPGA. Соответственно, в данной статье предлагается методика для проведения такого рода исследования.

Отметим, что процесс исследования является достаточно сложным в том отношении, что требует применения большого количества различных технических средств [3], поэтому в статье этот аспект рассматривается достаточно подробно.

Другим важным замечанием является то, что предлагаемая методика разрабатывалась таким образом, чтобы обеспечить возможность для ее использования в рамках параллельных систем. В [5] были рассмотрены вопросы, которые касались синтеза управляющих автоматов с использованием параллельных систем. Предложенные в [5] структуры и алгоритмы могут быть адаптированы для совместного использования с предложенной методикой.

ПОСТАНОВКА ЗАДАЧИ

Под понятием методики исследования в данной статье будем понимать совокупность взаимосвязанных составляющих. Первая составляющая – это множество видов деятельности, которые необходимы для проведения полного и достоверного исследования, а также порядок их выполнения. В свою очередь, вторая составляющая – это комплекс технических средств, требуемых для осуществления определенных ранее видов деятельности.

Поэтому разработка методики исследования предполагает решение двух подзадач:

1) декомпозиция процесса исследования на множество отдельных шагов (или же видов деятельности) с определением для каждого из них входных и выходных данных;

2) определение набора технических средств, необходимых для выполнения всех видов деятельности и разработка способа их организации в единый комплекс.

Немаловажным аспектом является и оценка возможностей применения предлагаемой методики в реальных условиях. Следовательно, актуальной является и третья подзадача – апробация возможностей практического применения методики исследования.

АЛГОРИТМ ИССЛЕДОВАНИЯ

В основу предлагаемой методики исследования положен алгоритм (рис. 1), который предполагает выполнение шести различных видов деятельности.

Вначале выполняется генерация исходной модели. Способ построения модели может быть разным в зависимости от того, какой именно метод оптимизации исследуется, но, базовая структура модели может оставаться неизменной. Исходя из этого, генератор модели должен позволять выполнить точную настройку процесса генерации по нескольким параметрам. В частности, это

может быть количество состояний автомата, количество символов входного и выходного алфавитов и т. д.

Далее выполняется оптимизация исходной модели. При этом следует учитывать, что в процессе оптимизации модель может расширяться за счет элементов, которые свойственны скорее цифровой технике, чем теории конечных автоматов. Оптимизированная модель будет представлять собой видоизмененную модель УА, которая взаимосвязана с набором моделей, описывающих необходимые функциональные узлы устройства.

Далее выполняется генерация формального описания моделей. Формальное описание может быть выполнено с использованием одного из языков описания аппаратуры, например, VHDL или Verilog. Данное описание должно соответствовать общепринятым рекомендациям для конкретного языка [6, 7]. Как минимальную достаточную оценку качества формального описания УА, можно рассматривать его корректное распознавание средствами синтеза.

Следующий вид деятельности – это верификация эквивалентности исходной модели ее формальному описанию, а также эквивалентности моделей. Доказательство эквивалентности моделей автоматов можно выполнить следующим образом. Примем, что два управляющих автомата имеют N_S различных состояний $s \in S$ и N_X сим-

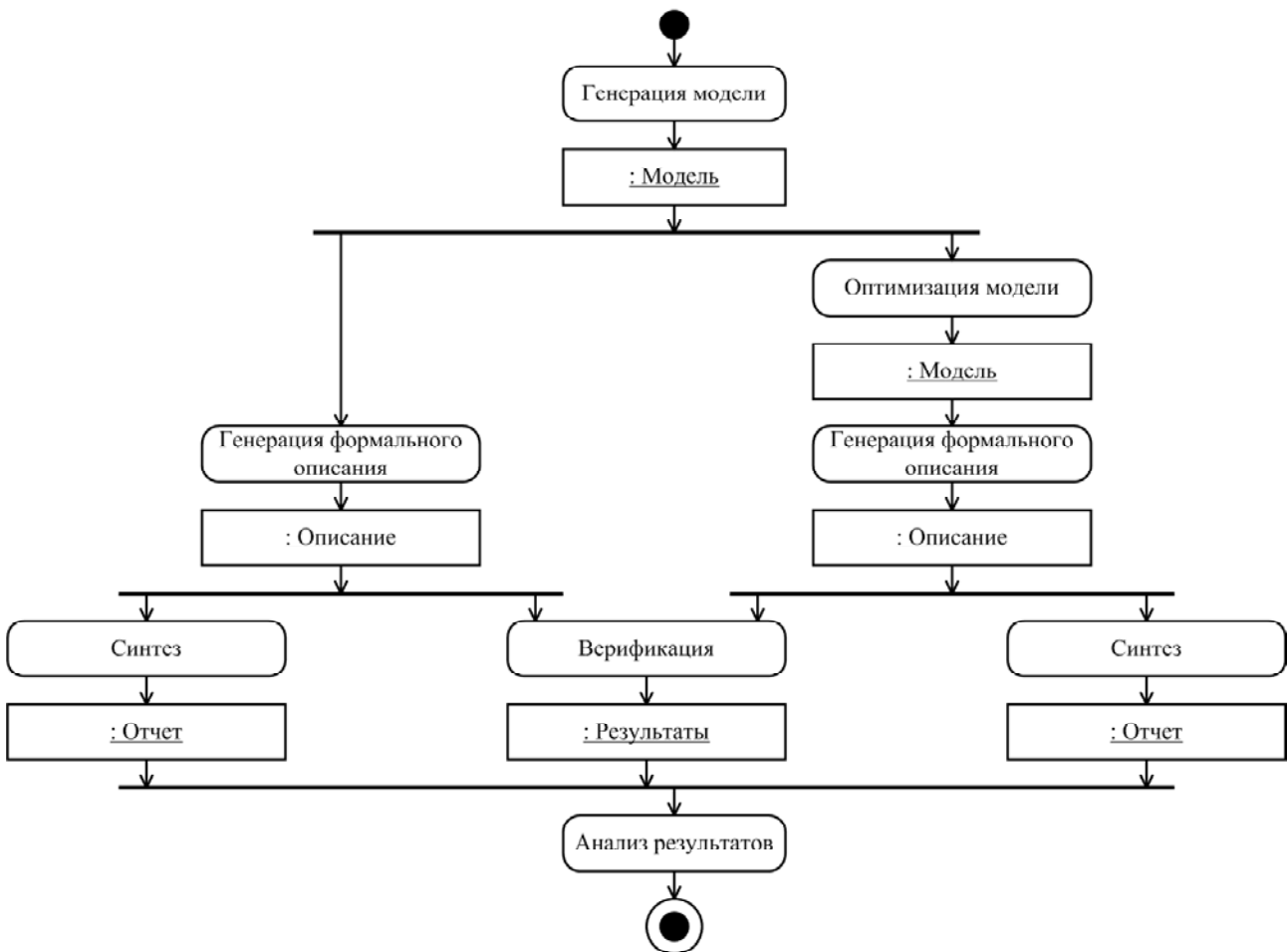


Рис. 1. Алгоритм исследования метода оптимизации управляющего автомата (диаграмма деятельности UML)

волов входного алфавита. В таком случае, для каждого состояния s_i существует не более чем 2^{N_x} исходящих переходов. Далее, примем, что существует возможность установки состояния s_i без необходимости изменения описания, что устраняет возможность привнесения дополнительной ошибки. Следовательно, для доказательства эквивалентности двух формальных описаний потребуются не более чем $N_S \times 2^{N_x}$ итераций. В этом случае актуальным является вопрос использования параллельных систем [5], поскольку речь идет о большом количестве однообразных действий.

Одновременно с верификацией необходимо выполнить синтез формальных описаний обеих моделей. На данном этапе важным аспектом является выбор элементной базы, т.е. конкретного семейства или определенной ПЛИС, и задание набора специальных параметров, в частности, способа кодирования состояний, глобальной стратегии оптимизации и т. д.

Последний вид деятельности – это анализ отчетов верификации и синтеза. Алгоритм на рис. 1 позволяет обработать только одну модель управляющего автомата. В то же время, для сбора необходимых статистических данных нужно выполнять обработку нескольких моделей, которые обладают схожими характеристиками. На данной стадии выполняется анализ всех полученных данных.

СТРУКТУРА КОМПЛЕКСА ПРОГРАММНЫХ СРЕДСТВ

Для выполнения предложенного алгоритма требуется сложный комплекс технических средств, который состоит из разнородных компонент (рис. 2).

Генератор исходной и оптимизированной моделей, как и генератор формального описания, могут реализовываться с использованием одного из современных языков программирования.

Основным критерием для выбора средства синтеза можно считать конкретную модель ПЛИС или же конкретное семейство ПЛИС, в базе которого выполняется исследование. В частности, если используется семейство ПЛИС Altera Stratix III, то необходима САПР Altera Quartus II, а для семейства Xilinx Spartan 6, соответственно, необходима САПР Xilinx ISE.

Средство верификации можно выбирать, исходя из рыночных предпочтений. Причем какая-либо связь между средством верификации и средством синтеза может отсутствовать. В частности, одной из наиболее распространенных сейчас САПР для моделирования, которую можно использовать и для верификации, является Mentor Graphics ModelSim.

Перечисленные средства по умолчанию не взаимосвязаны. Поэтому при разработке комплекса программного обеспечения требуется специальное средство для управления разными САПР, а также для осуществления сбора и обработки данных, что, в свою очередь, предполагает возможность создания оболочки-интерфейса вокруг алгоритма исследования, необходимой для сбора статистически значимых данных, что отмечалось в предыдущем разделе.

В настоящее время в качестве такого средства может выступать язык Tcl/Tk, который, для некоторых САПР (например, Altera Quartus II или Mentor Graphics ModelSim), является основным средством автоматизации.

Все перечисленные и показанные на рис. 2 компоненты являются обязательными. Однако, комплекс также может включать и дополнительные компоненты. В частности, на рис. 1 было показано, что определенные виды деятельности могут выполняться параллельно. Кроме того, отдельные виды деятельности, например, верификация, сами по себе обладают свойствами параллельных алгоритмов. Поэтому можно использовать компоненты, обеспечивающие ту или иную степень параллелизма [5].

Структура, приведенная на рис. 2, является масштабируемой, что позволяет выполнять исследование нескольких различных методов оптимизации. При этом алгоритм, приведенный на рис. 1, применяется к множеству пар моделей, в каждой из которых первой является исходная модель, а второй – одна из оптимизированных моделей.

ИССЛЕДОВАНИЕ МЕТОДА ЗАМЕЩЕНИЯ СИМВОЛОВ ВХОДНОГО АЛФАВИТА

Метод, основанный на замещении символов входного алфавита, описан в [1] и [2]. Его идея заключается в замещении исходного входного алфавита $X = \{x_1, \dots, x_N\}$

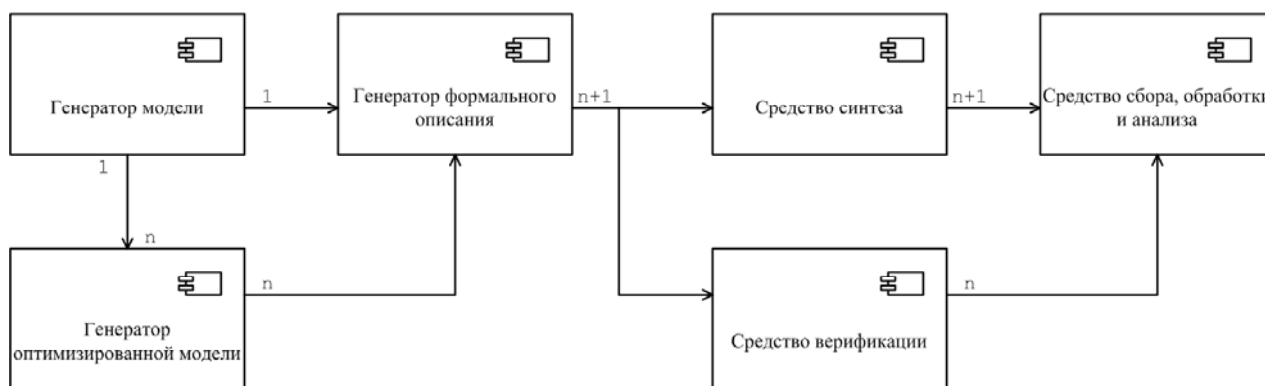


Рис. 2. Структура комплекса программного обеспечения

новым множеством $B = \{b_0, \dots, b_M\}$, которое включает меньшее количество символов (то есть $M \ll N$), определяемое числом символов, необходимым и достаточным для управления любым переходом внутри автомата. Данный метод ориентирован на уменьшение аппаратных затрат в логической схеме управляющего автомата.

Генератор модели УА был разработан так, чтобы обеспечить возможность изменения следующих параметров: N_X – количество символов входного алфавита, K_X – максимальное количество символов входного алфавита на переход, N_Y – количество символов выходного алфавита, K_Y – максимальное количество символов выходного алфавита на переход и N_S – количество состояний управляющего автомата.

Процесс оптимизации приводит к необходимости создания дополнительной модели коммутирования. Она может быть описана как ассоциативный массив, где ключом является состояние УА, а значениями – другие ассоциативные массивы, для которых ключом является символ входного алфавита, а значением – символ замещающего алфавита: $\langle s_i \rightarrow \langle x_j \rightarrow b_k \rangle \rangle$. Данная модель коммутирования является достаточно простой и дополняет модель автомата.

Верификация формального описания моделей выполнялась с использованием САПР Mentor Graphics ModelSim. Для работы с «внутренними» (т. е. недоступными непосредственно через внешний интерфейс) сигналами использовались сценарии на Tcl/Tk [8]. Хотя текущий стандарт VHDL [9] позволяет работать с такими сигналами, он поддерживается средствами моделирова-

ния лишь частично. Поэтому генерация тестовых сценариев не на том языке, который использовался для формального описания управляющего автомата, в данном случае является необходимой, хотя и нежелательной.

В качестве базиса были выбраны современные высокопроизводительные ПЛИС FPGA семейства Altera Stratix III [10]. Одна из основных архитектурных особенностей этих ПЛИС заключается в том, что они построены на генераторах функции от шести аргументов. Это дает возможность эффективно реализовывать достаточно сложные схемы коммутирования, характерные для исследуемого метода оптимизации.

В соответствии с элементарным базисом, в качестве САПР синтеза использовалась Altera Quartus II [11]. При этом для управления данной САПР, как и в случае с Mentor Graphics ModelSim, был использован язык Tcl/Tk.

Во избежание излишнего разнообразия используемых программных инструментов, язык Tcl/Tk использовался также для выполнения сбора, обработки и анализа отчетов САПР моделирования и синтеза.

Для получения статистически корректных данных для каждого набора характеристик выполнялась генерация десяти разных моделей управляющих автоматов. После выполнения исследования всех этих моделей результаты упорядочивались в зависимости от полученных выгод или потерь относительно аппаратных затрат. После этого наилучший и наихудший результаты отбрасывались, а остальные усреднялись, чтобы проследить общую тенденцию.

На рис. 3 приведены результаты, полученные для моделей, обладающих следующими характеристиками: $N_X = 10$, $K_X = 2$, $N_Y = 20$, $K_Y = 2$, $N = \{20, 22, \dots, 30\}$.

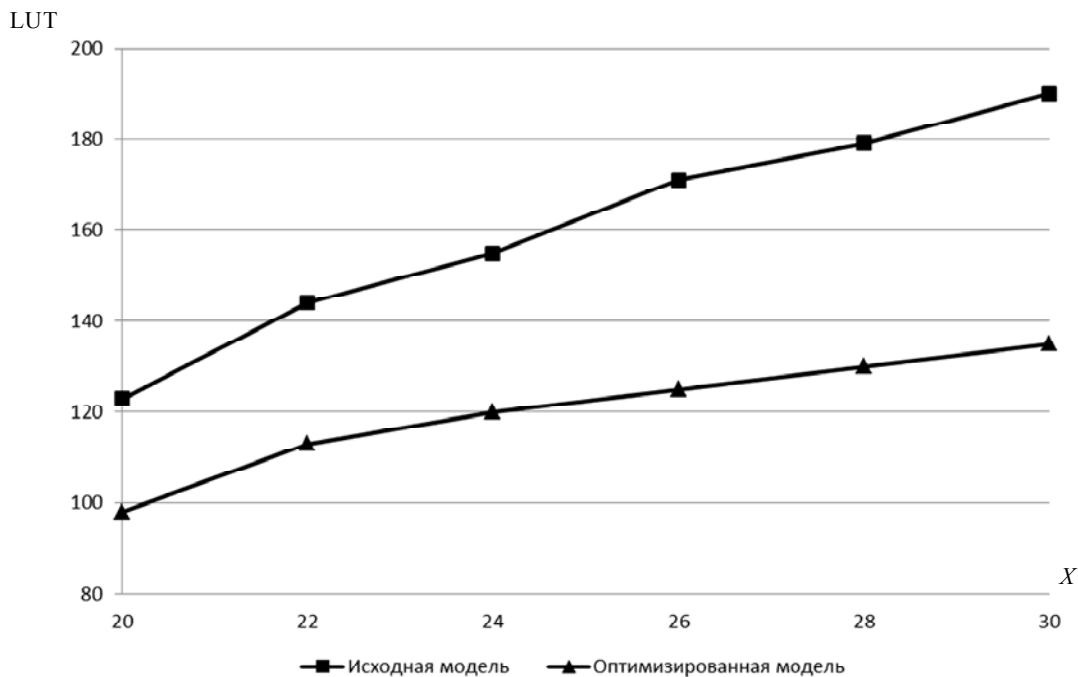


Рис. 3. Пример результатов исследования метода замещения символов входного алфавита для уменьшения количества необходимых для реализации LUT-элементов (параметры $N_X = 10$, $K_X = 2$, $N_Y = 20$, $K_Y = 2$, базис – Altera Stratix III)

Согласно этим результатам можно сделать вывод об эффективности применения исследуемого метода в базе ПЛИС семейства Stratix III. В данном примере аппаратные затраты определяются количеством необходимых LUT-элементов.

ЗАКЛЮЧЕНИЕ

В статье предложена методика, позволяющая выполнять исследование способов оптимизации схем управляющих автоматов, которая включает две составляющие: алгоритм проведения исследования и структуру комплекса технических средств, необходимых для его выполнения.

Алгоритм исследования представляет собой последовательность шагов, выполнение которых позволяет охватить все необходимые процессы, которые связаны, с одной стороны, с применением выбранного метода оптимизации, а с другой – с работой в базе ПЛИС FPGA. Предложенный алгоритм разработан таким образом, чтобы он мог быть реализован с использованием параллельных вычислительных систем, что является важным фактором.

Предложенная в статье структура комплекса технических средств охватывает полный набор программных средств, необходимых для выполнения рассмотренного алгоритма. При этом описание структуры включает достаточно детальный анализ реальных САПР, которые могут быть использованы. Также рассмотрены процессы автоматизации работы комплекса с использованием специализированных языков высокого уровня.

Кроме того, в данной статье показана апробация предложенной методики для одного из достаточно известных методов оптимизации, освещен ряд практических вопросов использования методики. Результаты апробации доказывают эффективность и актуальность применения предложенной в статье методики.

Последующими направлениями работы являются исследование различных способов оптимизации управляющих автоматов и изучение возможностей реализации предложенной методики с использованием параллельных вычислительных систем.

СПИСОК ЛИТЕРАТУРЫ

1. Баркалов, А. А. Синтез устройств управления на программируемых логических устройствах / А. А. Баркалов. – Донецк : ДонНТУ, 2002. – 262 с.
2. Баркалов, А. А. Синтез микропрограммных автоматов на заказных и программируемых СБИС / А. А. Баркалов, Л. А. Титаренко. – Донецк : ДонНТУ, Технопарк ДонНТУ УНИТЕХ, 2009. – 336 с.
3. Maxfield, C. FPGAs: Instant Access / C. Maxfield. – Oxford: Newnes, 2008. – 216 p.
4. Adamski, M. Design of Digital Systems and Devices / M. Adamski, A. Barkalov, M. Wegrzyn. – Berlin : Springer-Verlag, 2011. – 366 p.
5. Баркалов, А. А. Синтез управляющих автоматов с использованием распределенных и параллельных систем / А. А. Баркалов, И. Я. Зеленева, А. А. Гриценко // Радіоелектроніка, інформатика, управління. – 2010. – № 22. – С. 128–134.

6. Smith, D. J. HDL Chip Design: A Practical Guide for Designing, Synthesizing and Simulating ASICs and FPGAs Using VHDL or Verilog / D. J. Smith. – Austin : Doone Publications, 1998. – 448 p.
7. Chu, P.P. RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability / P. P. Chu. – New Jersey: John Wiley and Sons, 2006. – 694 p.
8. ModelSim ASIC and FPGA Design – Mentor Graphics [Электронный ресурс]. – Режим доступа: <http://www.mentor.com/products/fv/modelsim>.
9. Ashenden, P. J. VHDL 2008: Just the New Stuff (Systems on Silicon) / P. J. Ashenden, J. Lewis. – Burlington: Morgan Kaufmann, 2008. – 256 p.
10. Stratix III Device Handbook, Volume 1 [Электронный ресурс]. – Режим доступа: http://www.altera.com/literature/hb/stx3/stx3_siii5v1.pdf.
11. Quartus II Handbook Version 10.1 Volume 1: Design and Synthesis [Электронный ресурс]. – Режим доступа: http://www.altera.com/literature/hb/qts/quartusii_handbook.pdf.

Стаття надійшла до редакції 28.02.12.

Гриценко А. О., Зеленьова І. Я., Сіроштан С. Ю., Татолов С. Р.
МЕТОДИКА ДОСЛІДЖЕННЯ СПОСОБІВ ОПТИМІЗАЦІЇ
КЕРУЮЧИХ АВТОМАТІВ У БАЗИСІ ПЛІС FPGA

У статті запропонована методика, що дозволяє досліджувати відомі способи оптимізації керуючих автоматів для бази-су сучасних ПЛІС FPGA. Показана структура комплексу програмних засобів, необхідних для реалізації методики, а також представлені результати її апробації.

Ключові слова: ПЛІС, керуючий автомат, FPGA, синтез, верифікація, САПР.

Grytsenko A. A., Zelenyova I. J., Sirosthan S. Y., Tatolov E. R.
METHODOLOGY OF RESEARCH OF FSM
OPTIMIZATION METHODS FOR FPGA

Methodology of research of known FSM optimization methods for modern FPGAs is proposed. Such methodology consists of different FSM generation, optimization, synthesis, verification stages and can be implemented using parallel and distributed systems. That approach can lead to the efficient results because of methodology nature. The general idea is to figure out the necessary tools and algorithms to prove, that optimized model of FSM has the same behavior as initial model. The structure of standard and specialized software tools for methodology implementation is demonstrated and analyzed. The results of methodology approbation using well-known FSM optimization method targeting Altera FPGAs are pointed out.

Key words: FPGA, finite state machine, synthesis, verification, EDA.

REFERENCES

1. Barkalov A. A. Sintez ustrojstv upravleniya na programmiruemyx logicheskix ustrojstvax. Donetsk, DonNTU, 2002, 262 p.
2. Barkalov A. A. Sintez mikroprogrammnyx avtomatov na zakaznyx i programmiruemyx SBIS. Donetsk, DonNTU, Technopark DonNTU UNITEK, 2009, 336 p.
3. Maxfield C. FPGAs: Instant Access. Oxford, Newnes, 2008, 216 p.
4. Adamski M., Barkalov A., Wegrzyn M. Design of Digital Systems and Devices. Berlin, Springer-Verlag, 2011, 366 p.

5. Barkalov A.A., Zelenyova I.Ya., Gricenko A.A. Sintez upravlyayushhix avtomatov s ispol'zovaniem raspredelyonny'x i parallel'ny'x system, *Radioelektronika, informatyka, upravlinnia*, 2010, No. 22, pp. 128–134.
6. Smith D.J. HDL Chip Design: A Practical Guide for Designing, Synthesizing and Simulating ASICs and FPGAs Using VHDL or Verilog. Austin, Doone Publications, 1998, 448 p.
7. Chu P.P. RTL Hardware Design Using VHDL: Coding for Efficiency, Portability, and Scalability. New Jersey, John Wiley and Sons, 2006, 694 p.
8. ModelSim ASIC and FPGA Design – Mentor Graphics [Electronic Resource]. – Mode of access: <http://www.mentor.com/products/fv/modelsim>.
9. Ashenden P.J., Lewis J. VHDL 2008: Just the New Stuff (Systems on Silicon). Burlington, Morgan Kaufmann, 2008, 256 p.
10. Stratix III Device Handbook, Volume 1 [Electronic Resource]. – Mode of access: http://www.altera.com/literature/hb/stx3/stx3_siii5v1.pdf.
11. Quartus II Handbook Version 10.1 Volume 1: Design and Synthesis [Electronic Resource]. – Mode of access: http://www.altera.com/literature/hb/qts/quartusii_handbook.pdf.

УДК 004.031.4

Коломоец Г. П.¹, Пашков Д. В.²¹Канд. физ.-мат. наук, доцент Классического частного университета, г. Запорожье²Инженер-программист Классического частного университета, г. Запорожье

АГРЕГАТОР ЧИСЛЕННОСТИ КОНТИНГЕНТА УНИВЕРСИТЕТА

В статье описана используемая в Классическом частном университете, г. Запорожье программа-агрегатор численности контингента, которая осуществляет выборку информации из различных баз данных и представляет функциональный Web-интерфейс, позволяющий отслеживать изменение численности студентов и слушателей колледжа по различным направлениям/специальностям, курсам и формам обучения.

Ключевые слова: сервлет, Java Server Pages (JSP), Spring, база данных, Structured Query Language (SQL), JavaScript, jQuery, Web-страница.

ВВЕДЕНИЕ

Достаточно распространенной проблемой многих организаций, в том числе и Классического частного университета, г. Запорожье (КПУ), является отсутствие единой информационной системы управления (так называемая «кусочная автоматизация» [1]). В частности, в КПУ задачи приемной комиссии решает информационная система «Абитуриент», хранящая данные в СУБД *Microsoft SQL*, данные поступивших в ВУЗ абитуриентов экспортируются в базу данных СУБД *Firebird SQL*, с которой работает отдельная система отдела кадров. Кроме того, в обеих системах данные абитуриентов/студентов университета и колледжа находятся в различных базах данных. Поскольку мониторинг количества поступивших и обучающихся студентов является актуальным, руководством КПУ была поставлена задача разработки агрегатора с удобным интерфейсом, который выводил бы суммарные данные по контингенту студентов и слушателей колледжа из баз данных приемной комиссии и отдела кадров с группировкой по обучаемым направлениям/специальностям, курсам и формам обучения.

Цель статьи – представить решение задачи агрегирования данных различных систем в одном Web-ориентированном приложении, разработанном с использованием кроссплатформенных свободно распространяемых программных средств.

АРХИТЕКТУРА ПРИЛОЖЕНИЯ

Для обеспечения простого и удобного интерфейса с учетом обеспечения доступа с различных операцион-

ных систем, было принято решение о разработке Web-приложения на платформе Java с использованием технологий сервлетов, Java Server Pages (JSP) [2] и фреймворка Spring [3]. В качестве технологий доступа к различным базам данных были использованы технологии Java DataBase Connectivity (JDBC) [4] и Hibernate [3]. На рис. 1 приведена функциональная диаграмма разработанного приложения и показано место использованных технологий в ней.

Доступ к различным серверам и базам данных реализован с помощью сервлетов. Сервлет для работы с базами данных на основе XML-файла конфигурации Spring с параметрами подключения создает объект-источник данных (bean). Особенностью данного объекта-источника является использование разработанных авторами универсального метода подключения к различным базам данных и универсального метода, возвращающего данные из базы: либо в виде одномерной коллекции названий направлений/специальностей, либо в виде двумерной коллекции данных численности контингента по специальностям и курсам/формам обучения. Данные о параметрах подключения, имени, пароле и драйвере хранятся в XML-файле конфигурации Spring, что значительно упрощает поддержку системы. При необходимости изменения параметров подключения не нужно переписывать код Java-класса и компилировать его, достаточно всего лишь внести изменения в XML-файл.

Учитывая, что информация хранится в базах данных с различной структурой, были разработаны отдельные текстовые файлы с необходимыми SQL-запросами с параметрами, используемые в качестве параметров объек-