**Mean-square error of kernel regression on test function 4**

| | Without noise | 10 % noise |
|---|---|---|
| *Mean-square error without attribute 1* | 96,7 | 128,38 |
| Mean-square error without attribute 2 | 136,43 | 148,63 |
| Mean-square error without attribute 3 | 107,71 | 144,03 |
| Mean-square error without attribute 4 | 185,31 | 221,83 |
| Mean-square error without attribute 5 | 128,82 | 152,7 |
| Mean-square error without attribute 6 | 261,45 | 261,23 |
| Mean-square error without attribute 7 | 98,91 | 129,77 |
| Mean-square error without attribute 8 | 144,75 | 179,09 |
| Mean-square error without attribute 9 | 190,72 | 228,52 |
| Mean-square error with all attributes | 98,07 | 127,98 |

In addition we tested parallel version of our method for multiprocessor based on the parallelization of the genetic algorithm (fitness function calculation). We used multiprocessor with 2, 4, and 6 cores. And we got following values of the computing speed-up coefficients:

1) Configuration with 2 cores: 1,81–1,85;
2) Configuration with 4 cores: 3,20–3,76;
3) Configuration with 6 cores: 3,63–4,14.

So we can conclude that our method is appropriate for parallelization and using for multiprocessors.

## 5. Conclusions

So hybrid self-adjusted evolutionary algorithm is implemented for informative attribute selection. Reliability of its algorithm was experimented.

Genetic algorithm effectively solves optimization problem with bandwidth parameters in kernel regression. Hybrid self-adjusted algorithm solves problem of algorithm setting.

So hybrid self-adjusted genetic algorithm solves informative attribute selection problem on test functions effectively. Also this algorithm gives some data for analysis of information content of the attributes. The method is appropriate for parallelization.

## References

1. Aivazyan, S. A. [et al.] Applied statistics. Classification and reduction of dimensionality. Moscow, Finansy i statistika, 1989, 607 p.

2. Medvedev A. V. Non-parametrical systems of adaptation. Novosibirsk, Nauka, 1983. 174 p.

3. Goldberg D. E. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley, 1989.

4. Hall P., Li Q. and J. S. Racine, «Nonparametric Estimation of Regression Functions in the Presence of Irrelevant Regressors. *Review of Economics and Statistics*. 2007. 89. P. 784–789.

5. Gomez J. Self-adaptation of operator rates in evolutionary algorithms. *Proc. of Genetic and Evolutionary Computation Conference.* 2004. P. 1162–1173.

6. Volkova S., Sergienko R. B. Informative attributes selection in non-parametric regression estimation by making use of genetic algorithms. *Vestnik Tomskogo gosudarstvennogo universiteta. Ypravlenie, vichislitelnaya technika i informatica*. 2013. № 1 (22). P. 40–48.



УДК 004.6

# РАЗВИТИЕ МЕТОДОВ ЭКВИВАЛЕНТНОГО ПРЕОБРАЗОВАНИЯ ГЕРТ-СЕТЕЙ ДЛЯ АНАЛИЗА МУЛЬТИВЕРСИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Д. И. Ковалев, М. В. Сарамуд, М. В. Карасева, Ю. А. Нургалеева

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева
Российская Федерация, 660014, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31
E-mail: saramud@bk.ru

*В настоящее время практически все большие программные системы являются распределенными. Анализ мультиверсионного программного обеспечения (ПО) можно проводить, основываясь на распределенных системах обработки информации. На примере ГЕРТ-сети, моделирующей поведение системы Condor при расчете задачи с фиксированной продолжительностью в режимах без резервного копирования и миграции (режим Vanilla), показаны эквивалентные преобразования, позволяющие существенно упростить сеть и облегчить поиск петель. Приведены расчеты, позволяющие получить вероятностные характеристики приведенной сети. Описываются различные режимы работы системы Condor, преимущества распределенных гетерогенных систем обработки информации.*

*Ключевые слова: мультиверсионное программное обеспечение, ГЕРТ-сети, вероятностные характеристики.*

# DEVELOPMENT OF THE EQUIVALENT TRANSFORMATION OF GERT-NETWORKS METHODS FOR MULTIVERSION SOFTWARE ANALYSIS

D. I. Kovalev, M. V. Saramud, M. V. Karaseva, Yu. A. Nurgaleeva

Siberian State Aerospace University named after academician M. F. Reshetnev
31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660014, Russian Federation
E-mail: saramud@bk.ru

*Nowadays approximately all software is distributed. Analysis of multiversinal software can be done on the basis of distributed systems of data processing. This paper shows on GERT-network example that the equivalent transformations allow to simplify considerably the network and facilitate the search loops. GERT-networks simulate the behavior of Condor while the fixed-length modes computation without backup and migration (mode "Vanilla"). Calculations which allow to obtain probabilistic characteristics of the given network are given. The various modes of the operation Condor, the advantages of distributed heterogeneous information processing systems are described.*

*Keywords: multiversion software, GERT-network, probabilistic characteristics.*

Как уже не раз отмечалось, мультиверсионное программное обеспечения (ПО) обладает рядом особенностей, которые и позволяют классифицировать различное ПО по принципу, является ли оно мультиверсионным. Одной из особенностей является то, что область применения мультиверсионного программного обеспечения остается достаточно узкой в связи с тем, что повышение надежности при таком подходе идет за счет программной и ресурсной избыточности [1]. Вследствие этого большая часть таких программ используется для управления объектами, для которых надежность является ключевой характеристикой (например, космическая промышленность). Это обстоятельство, в свою очередь, затрудняет получение реальных технических характеристик таких систем [2]. С другой стороны, анализ мультиверсионного ПО можно проводить, основываясь на распределенных системах обработки информации. Ведутся исследования в этом направлении [3].

В настоящее время практически все большие программные системы являются распределенными. Распределенной называется такая система, в которой обработка информации сосредоточена не на одной вычислительной машине, а распределена между несколькими компьютерами. Зачастую, в таких системах компьютер пользователя управляется одной операционной системой (ОС), компьютер по сбору заявок на обработку запросов – второй, а компьютеры – узлы системы могут управляться третьей ОС, отличной от первых двух. В таком случае эти системы называются распределенными гетерогенными системами обработки информации.

Подобные системы могут быть построены при помощи готовых библиотек. Для примера возьмем проект Condor (http://www.cs.wisc.edu/condor). Condor позволяет использовать в составе единого кластера узлы, не только различающиеся в аппаратной части, но и работающие под разными операционными системами, что дает возможность использовать для вычислений существующую компьютерную технику и уже имеющиеся коммуникации, тем самым существенно удешевляя стоимость создания кластера [4].

Condor имеет несколько режимов запуска вычислительных задач: Standard, Vanilla, PVM, MPI, Globus, Java. Режимы PVM, MPI, Globus и Java – это поддержка совместимости для программ, написанных с использованием данных библиотек, поэтому мы не будем их рассматривать. Наиболее интересными с точки зрения мультиверсионного подхода являются режимы Standard (с резервным копированием) и Vanilla (без резервного копирования) [5].

В режиме Standard Condor делает контрольные точки с заданным интервалом. Контрольная точка – это «снимок» текущего состояния задачи. Если необходима миграция задачи (например, пользователь начал использование компьютера), то Condor создает образ контрольной точки, перемещает его на другую машину и возобновляет вычисления с места, где он остановился. Если вычисляющий узел завершил работу аварийно или нарушилась связь с узлом, то Condor размещает на другом узле последний сохраненный образ контрольной точки задачи и возобновляет вычисления. Таким образом, задача может непрерывно вычисляться в течение длительного периода времени. Данный режим соответствует режиму выполнения задачи с периодическим выполнением резервного копирования ее состояния.

Режим Vanilla позволяет запускать произвольное консольное приложение, но не позволяет осуществить миграцию и резервное копирование задачи.

Ниже представлены функциональные схемы ГЕРТ-сети, моделирующие поведение системы Condor при расчете задачи с фиксированной продолжительностью в режимах без резервного копирования и миграции (рис. 1).

ГЕРТ-сеть, изображенная на рис. 2, состоит из узлов, соответствующих началу и завершению каждой отдельной работы, и дуг, представляющих действительное время выполнения каждой работы. Перед тем как перейти к поиску петель данной сети, произведем некоторые преобразования, которые существенно упростят сеть [6]. Соответствующие замены показаны на рис. 3–5.

Таким образом, удалось получить эквивалентную сеть, показанную на рис. 6, для которой существенно упростился поиск петель.

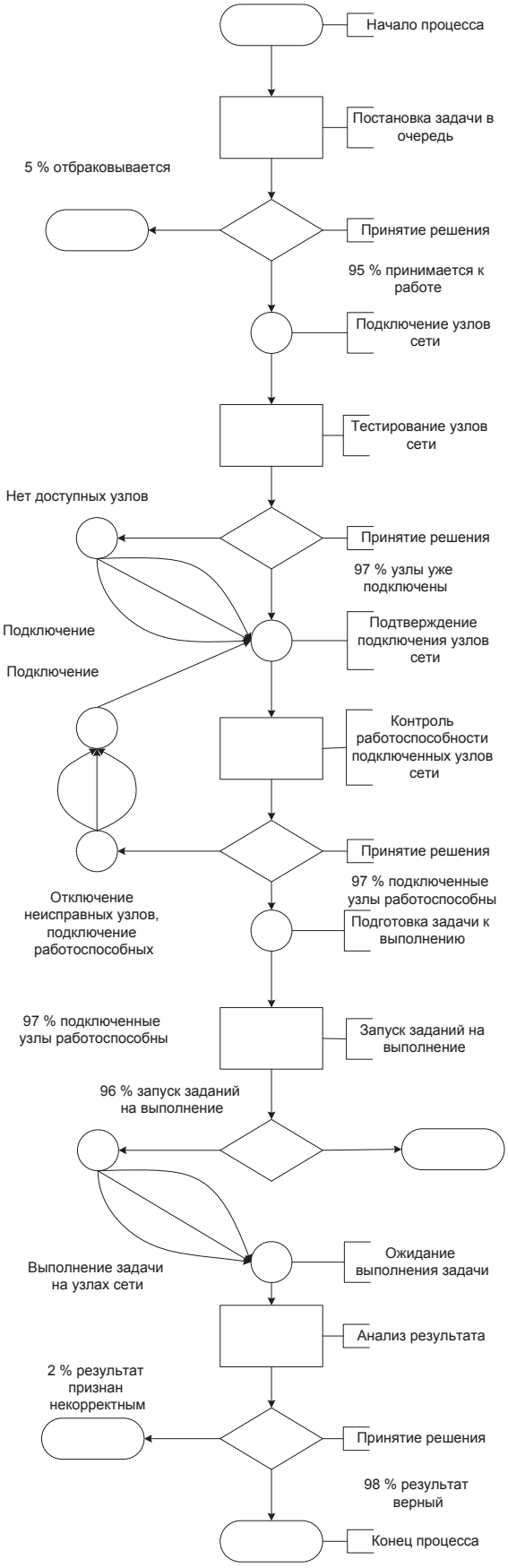


Рис. 1. Функциональная схема работы системы Condor в режиме Vanilla

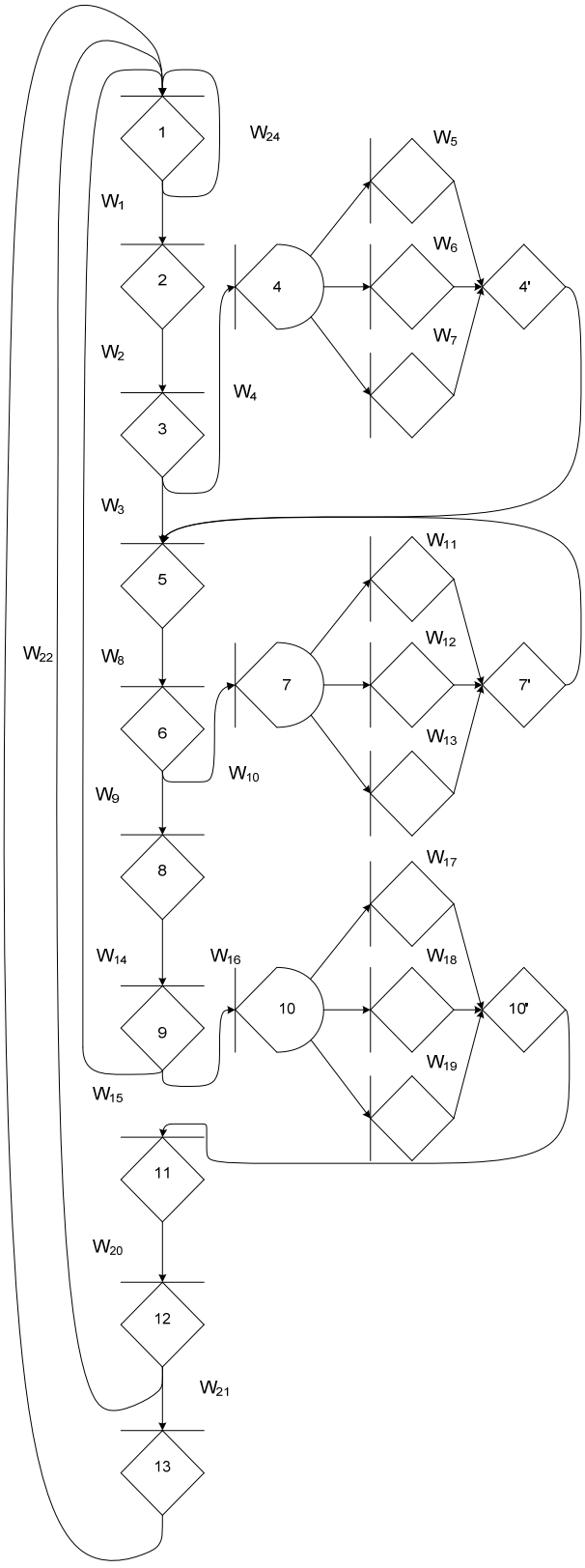


Рис. 2. ГЕРТ-сеть, описывающая работу системы Condor в режиме Vanilla

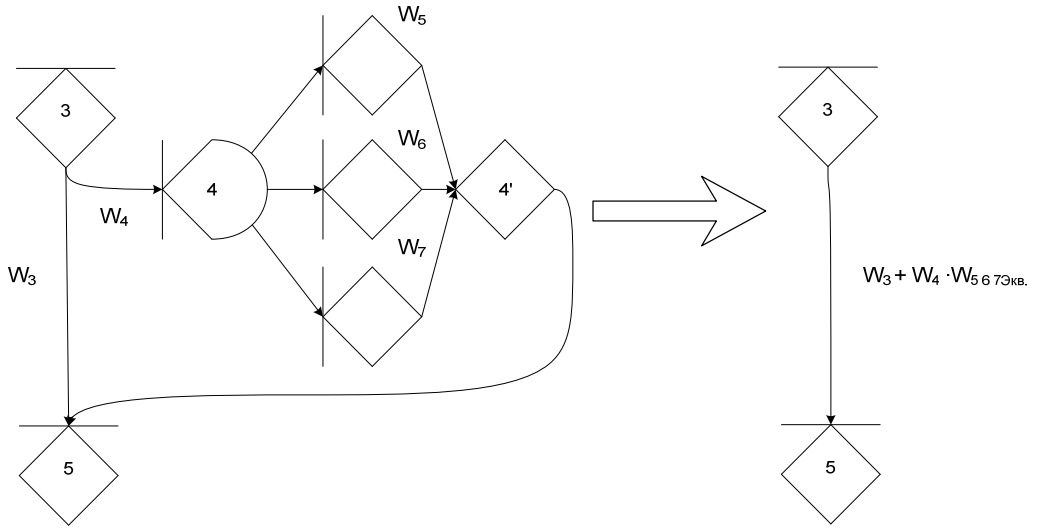


Рис. 3. Применение эквивалентных преобразований для узлов 3 и 5

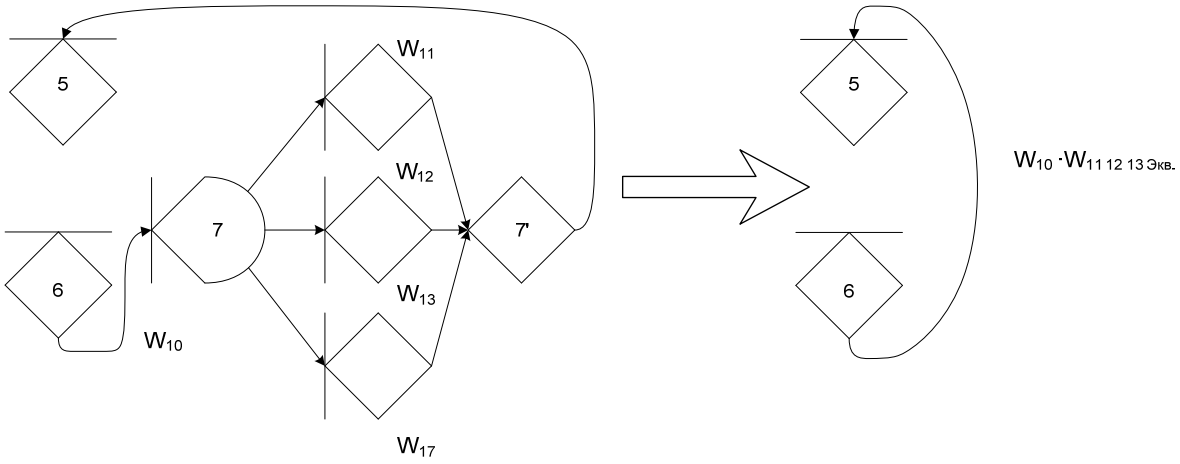


Рис. 4. Применение эквивалентных преобразований для узлов 5 и 6

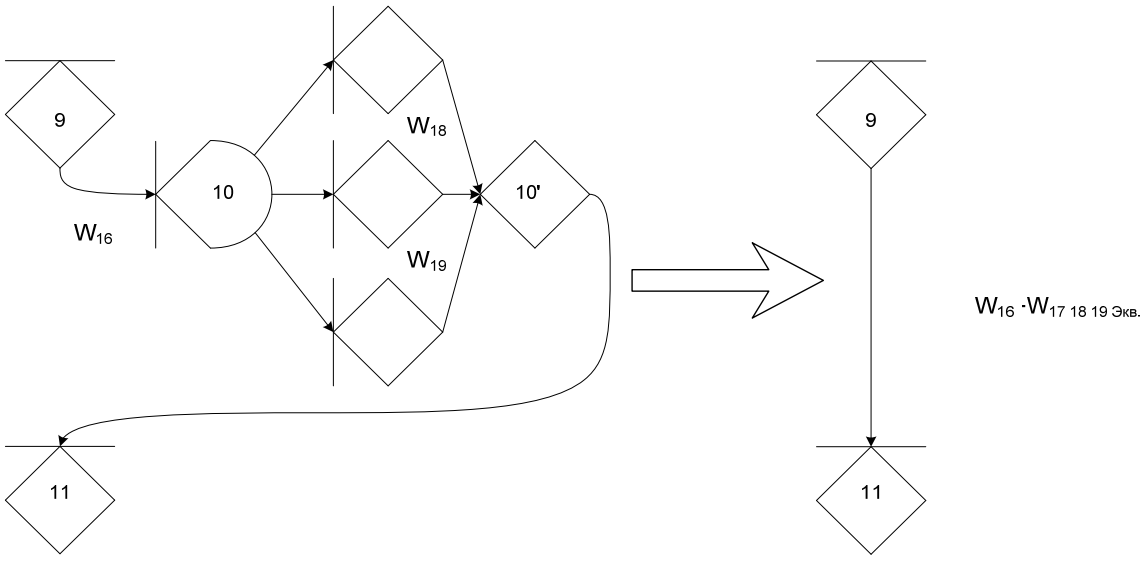


Рис. 5. Применение эквивалентных преобразований для узлов 9 и 11

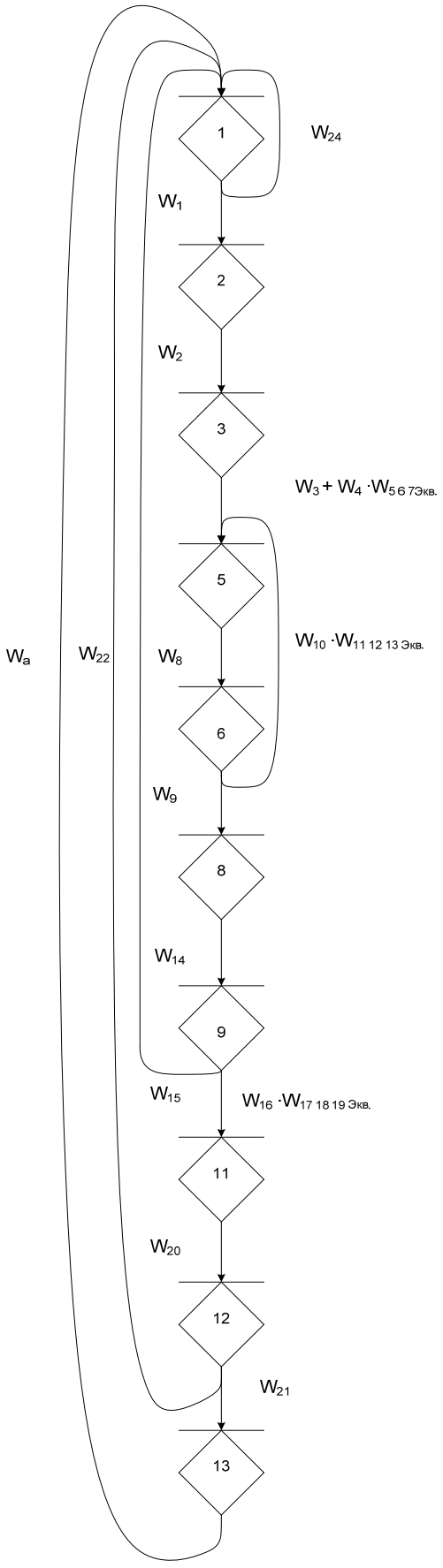


Рис. 6. Эквивалентная ГЕРТ-сеть, описывающая работу системы Condor в режиме Vanilla

По сети (рис. 6) определим соответствующие эквивалентные коэффициенты пропускания петель первого и второго порядка.

Петли первого порядка: $W_{24}$, $W_8 \cdot W_{10} \cdot W_{11,12,13\text{экв.}}$,

$W_1 \cdot W_2 \cdot W_8 \cdot W_9 \cdot W_{14} \cdot W_{15} \left( W_3 + W_4 \cdot W_{5,6,7\text{экв.}} \right)$,

$W_1 \cdot W_2 \cdot W_8 \cdot W_9 \cdot W_{14} \cdot W_{16} \cdot W_{17,18,19\text{экв.}} \cdot W_{20} \cdot W_{22}$ и

$W_1 \cdot W_2 \cdot W_8 \cdot W_9 \cdot W_{14} \cdot W_{16} \cdot W_{17,\,18,\,19\text{экв.}} \cdot W_{20} \cdot W_{21} \cdot \left( \frac{1}{W_E} \right)$.

Петли второго порядка: $W_8 \cdot W_{10} \cdot W_{11,12,13\text{экв.}} \cdot W_{24}$

Используя топологическое уравнение, получаем следующую эквивалентную $W$-функцию для ГЕРТ-сети, которая описывает работу системы без резервного копирования:

$$W_E(t) = \frac{W_1 W_2 W_8 W_9 W_{14} W_{16} W_{17,18,19\text{экв.}} W_{20} W_{21}}{1 - W_{24} - W_8 W_{10} W_{11,12,13\text{экв.}}} -$$

$$- \frac{W_1 W_2 W_8 W_9 W_{14} W_{16} W_{17,18,19\text{экв.}} W_{20} W_{21}}{W_1 W_2 W_8 W_9 W_{14} W_{15} \left( W_3 + W_4 W_{5,6,7\text{экв.}} \right)} -$$

$$- \frac{W_1 W_2 W_8 W_9 W_{14} W_{16} W_{17,18,19\text{экв.}} W_{20} W_{21}}{W_1 W_2 W_8 W_9 W_{14} W_{16} W_{17,18,19\text{экв.}} W_{20} W_{22}} +$$

$$+ \frac{W_1 W_2 W_8 W_9 W_{14} W_{16} W_{17,18,19\text{экв.}} W_{20} W_{21}}{W_8 W_{10} W_{11,12,13\text{экв.}} W_{24}}.$$

Используя полученные функции можно получить математическое ожидание и дисперсию.

Методы эквивалентного преобразования ГЕРТ-сетей позволяют упростить поиск петель и расчеты характеристик сети.

## Библиографические ссылки

1. Ковалёв П. В., Лайков А. Н., Гриценко С. Н. Определение надежности мультиверсионного программного обеспечения с использованием методов анализа сетей // Вестник СибГАУ : в 2 ч. Ч. 2. 2009. № 1 (22). С. 55–60.

2. Ковалёв П. В. ГЕРТ-сетевой анализ мультиверсионных архитектур программного обеспечения // Успехи современного естествознания. 2009. № 9. С. 161–164.

3. Доррер М. Г., Зырянов А. А. Прогноз динамики событийных моделей бизнес-процессов на основе GERT-сетей // Информация и связь. 2012. № 7. С. 124–127.

4. Ковалев И. В. Модели оценки времени выполнения задачи на кластере с последовательной и параллельной архитектурой обмена данными // Системы управления и информационные технологии. 2005. № 3(20). С. 58–62.

5. Дегтерев А. С., Письман Д. М. GERT-сетевой анализ времени выполнения задачи на неспециализированном гетерогенном кластере // Фундаментальные исследования. 2005. № 4. С. 79–80.

6. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей. М. : Мир, 1984.

**References**

1. Kovalev P. V., Laykov A. N., Gritsenko S. N. *Vestnik SibGAU.* 2009, no. 1(22) in 2 series. Series 2, p. 55–60.
2. Kovalev P. V. *Uspehi sovremennogo estestvoznaniya*. 2009, no. 9, p. 161–164.
3. Dorrer M. G., Ziryanov A. A. *Informatsiya I svyaz.* 2012, no. 7, p. 124–127.
4. Kovalev I. V. *Sistemi upravleniya I informatzionnie technologii.* 2005, no. 3 (20), p. 58–62.
5. Degterev A. S., Pisyman D. M. *Fundamentalynie issledovaniya*. 2005, no. 4, p. 79–80.
6. Fillips D., Garsia-Dias A. *Metodi analiza sete* (Methods for analyzing networks). Moscow, Mir, 1984.



УДК 004.056.55

# РЕАЛИЗАЦИЯ НА ПЛИС ШИФРА ЗАКРЕВСКОГО НА ОСНОВЕ ПЕРЕСТРАИВАЕМОГО АВТОМАТА

Д. С. Ковалев

ОАО «Информационные спутниковые системы» имени академика М. Ф. Решетнёва»
Российская Федерация, 662972, г. Железногорск Красноярского края, ул. Ленина, 52
E-mail: dmisk@hotmail.com

*Рассматривается задача проектирования быстродействующего компактного криптопроцессора. Предлагается решение данной задачи на базе конечно-автоматной модели. В качестве представителя класса автоматных шифров выбран шифр Закревского на основе перестраиваемого автомата. Представлена аппаратная реализация на ПЛИС Xilinx Spartan-3 данного шифра. Приведены значения ресурсоёмкости и производительности данной реализации. Произведено сравнение реализаций на ПЛИС шифра Закревского на основе перестраиваемого автомата, блочного шифра AES и поточных шифров – финалистов конкурса eSTREAM, рекомендованных для аппаратной реализации. Сделано заключение о возможности использования шифра Закревского на основе перестраиваемого автомата на практике.*

*Ключевые слова: криптопроцессор, ПЛИС, конечно-автоматная шифрсистема, шифр Закревского, перестраиваемый автомат.*

# FPGA IMPLEMENTATION OF THE ZAKREVSKIJ'S CIPHER BASED ON RECONFIGURABLE FSM

D. S. Kovalev

JSC “Information satellite system” named after academician M. F. Reshetnev”
52, Lenin str., Zheleznogorsk, Krasnoyarsk region, 662972, Russian Federation
E-mail: dmisk@hotmail.com

*The problem of the compact high-throughput cryptoprocessor design is considered. This problem solution based on the Finite State Machine model is offered. Zakrevskij's Cipher Based on Reconfigurable FSM is chosen as a representative of the finite automata ciphers class. FPGA Xilinx Spartan-3 hardware implementation of this cipher is presented. Values of the hardware resource using and throughput of this implementation are given. The FPGA implementation comparison of Zakrevskij's Cipher Based on Reconfigurable FSM, block cipher AES and hardware-oriented stream ciphers became eSTREAM competition finalists was made. The conclusion about Zakrevskij's Cipher Based on Reconfigurable FSM practice using possibility is made.*

*Keywords: cryptoprocessor, FPGA, finite automata cryptosystem, Zakrevskij’s cipher, reconfigurable finite state machine.*

В настоящее время существует высокая потребность в создании быстродействующих компактных криптопроцессоров, которые используются для снижения вычислительной нагрузки основного процессора, в средствах автоматического регулирования и управления техпроцессами, в телекоммуникационном оборудовании, так же как аппаратные модули доверенной загрузки операционной системы и пр.

Криптопроцессоры, как правило, являются устройствами, в которых реализованы криптографические алгоритмы различного назначения (симметричное/ асимметричное шифрование, электронная цифровая