

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 21
2015
№ 10

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

25 ЛЕТ НИИСИ РАН

- Бетелин В. Б., Галатенко В. А., Костюхин К. А.** Контролируемое выполнение приложений на многопроцессорных платформах 723
- Бобков С. Г., Сердин О. В.** Проблемы унификации микросхем космического применения на примере международного проекта космического телескопа "ГАММА-400" 728
- Барских М. Е., Бобков С. Г.** Исследование влияния динамического предсказания ветвлений на производительность перспективных микропроцессоров НИИСИ РАН 736

СТРУКТУРНЫЙ СИНТЕЗ И ПРИНЯТИЕ РЕШЕНИЙ

- Вишнеков А. В., Иванова Е. М.** К вопросу об автоматизации процедур определения победителей конкурсов научно-технических проектов и стартапов . . . 743

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

- Стемпковский А. Л., Тельпухов Д. В., Соловьев Р. А., Мячиков М. В.** Повышение отказоустойчивости логических схем с использованием нестандартных мажоритарных элементов. 749

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Змеев Д. Н., Климов А. В., Левченко Н. Н., Окунев А. С., Стемпковский А. Л.** Эмуляция аппаратно-программных средств параллельной потоковой вычислительной системы "Буран". 757

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

В ОРГАНИЗАЦИОННЫХ И СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ

- Спивак С. И., Кантор О. Г.** Чебышевское приближение в задачах эконометрического моделирования. 764

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

- Ракитин В. В., Русаков С. Г.** Генераторные схемы на мемристорных элементах . . . 772

Журнал в журнале

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Барский А. Б.** Обучаемые и самообучающиеся системы распознавания, управления и принятия решений на логических нейронных сетях. 783
- Куравский Л. С., Мармалюк П. А., Юрьев Г. А., Думин П. Н.** Методы численной идентификации марковских моделей и их сравнительный анализ 792

Главный редактор:
СТЕМПКОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:
ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:
БЫЧКОВ И. В., акад. РАН, д. т. н.
ЖУРАВЛЕВ Ю. И.,
акад. РАН, д. ф.-м. н., проф.
КУЛЕШОВ А. П.,
акад. РАН, д. т. н., проф.
ПОПКОВ Ю. С.,
чл.-корр. РАН, д. т. н., проф.
РУСАКОВ С. Г.,
чл.-корр. РАН, д. т. н., проф.
РЯБОВ Г. Г.,
чл.-корр. РАН, д. т. н., проф.
СОЙФЕР В. А.,
чл.-корр. РАН, д. т. н., проф.
СОКОЛОВ И. А., акад.
РАН, д. т. н., проф.
СУЕТИН Н. В., д. ф.-м. н., проф.
ЧАПЛЫГИН Ю. А.,
чл.-корр. РАН, д. т. н., проф.
ШАХНОВ В. А.,
чл.-корр. РАН, д. т. н., проф.
ШОКИН Ю. И.,
акад. РАН, д. т. н., проф.
ЮСУПОВ Р. М.,
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:
АВДОШИН С. М., к. т. н., доц.
АНТОНОВ Б. И.
БАРСКИЙ А. Б., д. т. н., проф.
ВАСЕНИН В. А., д. ф.-м. н., проф.
ВИШНЕКОВ А. В., д. т. н., проф.
ГАЛУШКИН А. И., д. т. н., проф.
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.
ДОМРАЧЕВ В. Г., д. т. н., проф.
ЗАБОРОВСКИЙ В. С., д. т. н., проф.
ЗАГИДУЛЛИН Р. Ш., к. т. н., доц.
ЗАРУБИН В. С., д. т. н., проф.
КАРПЕНКО А. П., д. ф.-м. н., проф.
КОЛИН К. К., д. т. н., проф.
КУЛАГИН В. П., д. т. н., проф.
КУРЕЙЧИК В. М., д. т. н., проф.
КУХАРЕНКО Б. Г., к. ф.-м. н., доц.
ЛЬВОВИЧ Я. Е., д. т. н., проф.
МИХАЙЛОВ Б. М., д. т. н., проф.
НЕЧАЕВ В. В., к. т. н., проф.
ПОЛЕЩУК О. М., д. т. н., проф.
СОКОЛОВ Б. В., д. т. н., проф.
УСКОВ В. Л., к. т. н. (США)
ТИМОНИНА Е. Е., д. т. н., проф.
УСКОВ В. Л., к. т. н. (США)
ФОМИЧЕВ В. А., д. т. н., проф.
ЧЕРМОШЕНЦЕВ С. Ф., д. т. н., проф.
ШИЛОВ В. В., к. т. н., доц.

Редакция:
БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

INFORMATION TECHNOLOGIES

INFORMACIONNYYE TEHNOLOGII

Vol. 21
2015
No. 10

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

25 YEARS NIISI RAS

- Betelin V. B., Galatenko V. A., Kostyukhin K. A.** Controlled Execution of Tightly-Coupled Multiprocessor Systems 723
- Bobkov S. G., Serdin O. V.** Unification of Chip Space Application on the Example of International Project Space Observatory GAMMA-400 728
- Barskikh M. E., Bobkov S. G.** Investigation of the Influence of Dynamic Branch Prediction to Perspective Microprocessors Performance from NIISI RAS 736

STRUCTURAL SYNTHESIS AND DECISION MAKING

- Vishnekov A. V., Ivanova E. M.** To the Question of Winners Definition Procedures Automation for the Scientific and Technical Projects and Startups Competition . . . 743

INTELLIGENT SYSTEMS AND TECHNOLOGIES

- Stempkovskiy A. L., Telpukhov D. V., Solovyev R. A., Myachikov M. V.** Improving the Fault Tolerance of Logic Circuits Using Unconventional Majority Voters . . . 749

SOFTWARE ENGINEERING

- Zmejev D. N., Klimov A. V., Levchenko N. N., Okunev A. S., Stempkovsky A. L.** Emulation on Hardware and Software of the Parallel Computing Streaming System "Buran". 757

INFORMATION TECHNOLOGIES IN THE ORGANIZATIONAL AND SOCIO-ECONOMIC SYSTEMS

- Spivak S. I., Kantor O. G.** Chebyshev Approximation in Problems of Econometric Modeling 764

APPLIED INFORMATION TECHNOLOGIES

- Rakitin V. V., Rusakov S. G.** Memristor-Based Oscillator Circuits 772

Journal-in-journal NEUROTECHNOLOGIES

- Barsky A. B.** Learned and Self-Learning Recognition, Management and Decision-Making Systems on Base of the Logical Neural Networks 783
- Kuravsky L. S., Marmalyuk P. A., Yuryev G. A., Dumin P. N.** Methods of Numerical Identification for Markov Models and Their Comparative Analysis 792

Editor-in-Chief:

Stempkovsky A. L., Member of RAS,
Dr. Sci. (Tech.), Prof.

Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.
Filimonov N. B., Dr. Sci. (Tech.), Prof.

Chairman:

Bychkov I. V., Member of RAS,
Dr. Sci. (Tech.), Prof.
Zhuravljov Yu. I., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Kuleshov A. P., Member of RAS,
Dr. Sci. (Tech.), Prof.
Popkov Yu. S., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Rusakov S. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Soifer V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Sokolov I. A., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Suetin N. V.,
Dr. Sci. (Phys.-Math.), Prof.
Chaplygin Yu. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shakhnov V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shokin Yu. I., Member of RAS,
Dr. Sci. (Tech.), Prof.
Yusupov R. M., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.
Antonov B. I.
Barsky A. B., Dr. Sci. (Tech.), Prof.
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.
Vishnekov A. V., Dr. Sci. (Tech.), Prof.
Galushkin A. I., Dr. Sci. (Tech.), Prof.
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.
Domrachev V. G., Dr. Sci. (Tech.), Prof.
Zaborovsky V. S., Dr. Sci. (Tech.), Prof.
Zagidullin R. Sh., Cand. Sci. (Tech.), Ass. Prof.
Zarubin V. S., Dr. Sci. (Tech.), Prof.
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.
Kolin K. K., Dr. Sci. (Tech.)
Kulagin V. P., Dr. Sci. (Tech.), Prof.
Kureichik V. M., Dr. Sci. (Tech.), Prof.
Kukhareno B. G., Cand. Sci. (Phys.-Math.)
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.
Mikhailov B. M., Dr. Sci. (Tech.), Prof.
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.
Poleschuk O. M., Dr. Sci. (Tech.), Prof.
Sokolov B. V., Dr. Sci. (Tech.)
Timonina E. E., Dr. Sci. (Tech.), Prof.
Uskov V. L. (USA), Dr. Sci. (Tech.)
Fomichev V. A., Dr. Sci. (Tech.), Prof.
Chermoshentsev S. F., Dr. Sci. (Tech.), Prof.
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

Editors:

Bezmenova M. Yu.
Grigorin-Ryabova E. V.
Lysenko A. V.
Chugunova A. V.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

УДК 004.3.06

В. Б. Бетелин, акад. РАН, директор, e-mail: betelin@niisi.msk.ru,

В. А. Галатенко, д-р физ.-мат. наук, ст. науч. сотр., зав. сектором, e-mail: galat@niisi.msk.ru,

К. А. Костюхин, канд. физ.-мат. наук, ст. науч. сотр., e-mail: kost@niisi.msk.ru

НИИ системных исследований РАН, Москва

Контролируемое выполнение приложений на многопроцессорных платформах

Разработка качественных сложных аппаратно-программных комплексов — длительный, трудоемкий процесс. Считается, что около половины времени уходит на отладку. Переход на многоядерные процессорные архитектуры сделал использование сильносвязанных систем нормой. Это делает отладку таких систем еще более важной и, одновременно, более сложной. Рассматривается предложенная авторами концепция контролируемого выполнения и ее применение к отладке сильносвязанных многопроцессорных комплексов.

Ключевые слова: контролируемое выполнение, отладка, мониторинг, трассировка, сильносвязанные системы

Введение

Специализированные сильносвязанные аппаратно-программные комплексы, предназначенные для решения ответственных задач, таких как обработка потока данных в реальном времени, в общем случае строятся из множества компонентов [1].

Основная особенность сильносвязанных систем — использование сложных, зачастую асинхронных, взаимодействий между компонентами системы. Эта особенность влияет на подходы к отладке систем, определяет выбор инструментов и методов отладки. Сложность отладки также обусловлена числом компонентов сильносвязанных систем, часть из которых могут быть аппаратными. Традиционный набор инструментов отладки в широком смысле (интерактивный отладчик, трассировщик, библиотеки самоконтроля, воспроизведения и отладки производительности) не теряют свою актуальность, но варианты их применения изменяются.

Определение концепции контролируемого выполнения

Аппаратно-программные комплексы создаются для решения определенных задач, достижения заданных целей. Для краткости будем говорить, что у каждого комплекса есть *миссия*.

Контролируемое выполнение — это процесс функционирования аппаратно-программного комплекса, при котором комплекс имеет возможность выпол-

нить свою миссию, несмотря на возможное проявление ошибок, атаки и отказы.

Традиционно отладку и мониторинг относят к этапам разработки, тестирования и отладки систем. Широкое распространение программного обеспечения с открытыми исходными текстами, по крайней мере в принципиальном плане, дает возможность распространить отладку и на этап эксплуатации. Разумеется, чтобы подобное решение стало разумным, необходимо изменение взгляда на эксплуатационную документацию к программным системам, а также на информацию, доступную во время выполнения. Их следует дополнить, предоставив пользователям и отладчику достаточно сведений для анализа нештатных ситуаций и их разрешения.

Для сложных ответственных сильносвязанных комплексов целесообразно постоянно контролировать характер их функционирования, по возможности выявляя и ликвидируя проблемы на ранних стадиях возникновения. Поведение таких комплексов определяется не только программными компонентами, но и наличием доступных ресурсов, сбоеустойчивостью аппаратных компонентов и т. п. Если не располагать регистрационной информацией, то на практике не удастся воспроизвести ситуацию, приведшую к ошибке и, соответственно, не удастся организовать отладку.

Применение средств самоконтроля программ можно рассматривать как распространение объектно-ориентированного подхода на контролируемое выполнение комплексов. Программы в таком

случае трактуются не как пассивные объекты, к которым внешним образом применяются отладочные и управляющие действия с внешней же интерпретацией семантики выполнения, а как активные сущности, самостоятельно контролирующие корректность своего функционирования.

Средства аппаратной отладки

ЕJTAG. Традиционным способом аппаратной отладки сильносвязанных комплексов является отладка через специальный инженерный порт. В тех комплексах, с которыми имеют дело авторы статьи, для этих целей используется интерфейс EJTAG [2].

EJTAG (Enhanced JTAG) — это спецификация аппаратно-программной подсистемы для семейства процессоров с ядрами MIPS. EJTAG использует инфраструктуру IEEE 1149.1 JTAG [3] в качестве интерфейса доступа к данным и командам и расширяет семейство инструкций ядра MIPS, а также набор компонент ядра. Таким образом создается унифицированная архитектура для отладки встраиваемых систем. Спецификации EJTAG постоянно обновляются.

Первоначально в основу отладочной архитектуры MIPS легла RISC-философия, ориентирующаяся на минимизацию аппаратной составляющей. Первые ядра MIPS определяли небольшой набор примитивов отладки — инструкцию прерывания, порождающую исключение, инструкции проверки значений регистров, пару необязательных регистров слежения за памятью, а также специальный режим устройства управления памятью, позволяющий реализовать защиту от изменения. Все эти механизмы требовали поддержки программного обеспечения, выполняющегося на процессоре, — операционной системы или некоторого монитора, позволяющего вставлять точки прерывания и передавать данные пользователю. Кроме того, внешние анализаторы логики и эмуляторы контуров могли управлять выполнением приложений. Эта модель была достаточно эффективной, но при появлении сложных многоядерных конфигураций с несколькими ядрами на одном чипе, использование внешних интерфейсов процессора стало значительно более сложным. Одним из препятствий стала отладка приложений, выполняющихся из неизменяемой памяти. Другим препятствием стала необходимость наличия специального отладочного порта, что значительно затрудняет разработку встраиваемых систем, увеличивая их стоимость. Важно обеспечить отладку приложения в полевых условиях. Отладка с использованием интерфейса EJTAG позволила решить эти проблемы.

Процессор с поддержкой EJTAG может быть встроен в цепь устройств и отлажен с использованием внешнего оборудования. EJTAG использует стандартный интерфейс для доступа к TAP. Для управления процессором используются регистры

Регистры EJTAG

IMPCODE	Регистр, содержащий информацию о процессоре и реализованных в нем функциях EJTAG
ADDRESS	Регистр для доступа к шине адреса
DATA	Регистр для доступа к шине данных
CONTROL	Регистр управления процессором
ALL	Регистр для одновременного доступа к шинам адреса, данных и управления
EJTAGBOOT	Регистр для установки режима отладки после перезагрузки
NORMALBOOT	Регистр для обычной перезагрузки
FASTDATA	Регистр для ускоренного доступа к данным
IMPCODE	Регистр, содержащий информацию о процессоре и реализованных в нем функциях EJTAG
TCBCONTROL[A-E]	Регистры для доступа к блоку трассировки
TCBDATA	Регистр для доступа к данным трассировки
PCSAMPLE	Регистр для профилирования
FDC	Регистр для доступа к каналу быстрой отладки

контроллера TAP, дополняющие набор стандартных регистров IDCODE, BYPASS и др. Эти регистры приведены в таблице.

Спецификация EJTAG описывает следующие возможности.

Отладочное исключение и отладочный режим.

Для того чтобы предоставить доступ к состоянию процессора в любой момент выполнения приложения, в спецификации EJTAG описывается отладочное исключение, после которого процессор переходит в специальный отладочный режим. В этом режиме возможен свободный доступ к сопроцессору, памяти и другим устройствам. Исключения по доступу к адресу и прерывания маскируются. Обработчик отладочного исключения предоставляется отладочной системой или приложением, если это предусмотрено разработчиком. Если это необходимо, разработчик может при наступлении отладочного исключения послать сигнал о нем внешним аппаратным модулям, заставляя их приостановить работу на время отладки.

Внешняя память EJTAG. EJTAG позволяет процессору MIPS в отладочном режиме выполнять инструкции, поступающие из TAP интерфейса. Для этого память EJTAG в специально выделенном сегменте преобразуется в операции над интерфейсом TAP. И данные, и инструкции становятся доступными на инструментальной платформе, что позволяет отлаживать программы в постоянной памяти. Коммуникация с агентом отладки осуществляется посредством регистра CONTROL, бит которого говорит об ожидании процессора. После этого адрес транзакции выставляется в регистре ADDRESS, а данные — в регистре DATA. Бит в ре-

гистре управления обновляется, чтобы сигнализировать, что транзакция совершена.

С частотой 40 МГц это занимает порядка 5 мкс, а скорость передачи данных составляет 800 кбайт/с. Оптимизация передачи может быть осуществлена посредством использования программных методов, таких как предсказание следующего адреса, или с помощью регистра FASTDATA.

Инструкции останова. Спецификации EJTAG определяют новую инструкцию SDBBP, отличающуюся от инструкции BREAK ядер MIPS32 и MIPS64 тем, что ее исключение переводит процессор в отладочный режим и позволяет выполнить обработчик из внешней памяти EJTAG.

EJTAG определяет различные типы аппаратных точек останова. Исключение порождается до того, как действие приводит к изменению состояния процессора или памяти, например, прерывание по выполнению по адресу происходит после того, как инструкция будет извлечена, но до того, как она будет выполнена. Прерывание по записи в память выполняется до того, как запись будет проведена. Аппаратные точки останова срабатывают без изменения памяти и поэтому обладают преимуществом перед программными.

EJTAG определяет два типа простых точек останова:

- останов по выполнению инструкции по заданному виртуальному адресу;
- останов по доступу к данным, причем срабатывающий при некотором значении записываемых/считываемых данных.

Возможна реализация до 15 точек останова различных типов.

Начиная с версии 4.00 спецификации EJTAG определяют сложные точки останова различных типов, например:

- инструкции со счетчиками срабатывания;
- инструкции с условием останова по адресу выполнения и данным одновременно;
- точки останова для заданных процессов;
- точки останова, зависящие от других точек останова;
- точки останова по таймеру.

Точки останова могут сигнализировать о начале трассировки, работа которой определяется спецификацией MIPS PDTrace [4].

Режимы трассировки и профилирования

Спецификации EJTAG также содержат разделы, посвященные аппаратной трассировке PDTrace и аппаратному профилированию. Аппаратная трассировка позволяет передавать данные о выполнении приложения по специальному внешнему каналу на инструментальную машину. Аппаратная трассировка настраивается с помощью регистров TAP.

Поддержка аппаратного профилирования заключается в периодическом сохранении в память

инструментальной машины текущей информации о выполнении приложения, такой как адреса инструкций, промахи в кэш-память или записываемые данные в регистр TAP PCSAMPLE, посредством которого она становится доступна на инструментальной платформе.

Аппаратные компоненты

В общем случае аппаратная инфраструктура EJTAG обладает следующими свойствами:

- наличие режима пошагового выполнения;
- наличие порта TAP;
- наличие точек останова, в том числе точек слежения;
- наличие управляющих регистров;
- наличие регистров профилирования;
- наличие канала быстрой передачи данных.

Расширения для многоядерной отладки. Спецификации MIPS MT ASE определяют процессор, состоящий из нескольких виртуальных модулей VPE (виртуальный процессорный элемент). С точки зрения аппаратуры и приложений такой процессор не отличается от набора нескольких процессоров. EJTAG может поддерживать все VPE по отдельности или модуль целиком. В первом случае каждое ядро должно иметь свой контроллер TAP и все они должны быть связаны в единую цепь, как определяет спецификация JTAG 1147.1. При этом значительная часть аппаратной составляющей не разделяется между модулями, каждый имеет свой отладочный регистр, регистр точек останова и т. д. Во втором случае ядра могут иметь один регистр отладки, поддерживать одновременное пошаговое выполнение и другие совместные отладочные действия. Версии спецификаций EJTAG определяют необходимые в этом случае изменения в значениях регистров.

Использование средств аппаратной отладки позволяет эффективно отлаживать сильносвязанные комплексы, поскольку имеется возможность практически одновременной остановки всех компонентов комплекса. В этом случае уменьшается риск нарушения их синхронизации.

Отладка аппаратных конфигураций. Современная постановка задачи отладки включает в себя такие аспекты, как отладка специфических аппаратных конфигураций, включающих в себя процессоры цифровой обработки сигналов или графические процессоры (см. Total Recall [5], NUDA [6]). Для отладки сложных сетей на кристалле предназначенные инструменты, описанные в работе [7].

Средства программной отладки и мониторинга

Интерактивная отладка. Традиционно под отладкой понимается в первую очередь интерактивная отладка. В то же время чем сложнее отлаживаемая система, тем менее эффективной оказывается по-

добная отладка. Слишком много факторов приходится принимать во внимание, слишком много данных необходимо проанализировать, чтобы ответить, например, на простейшие, но и важнейшие вопросы: случились ли уже в системе ошибки, и каковы их первопричины.

Вероятно, наиболее популярным инструментом интерактивной отладки является GDB [8]. На момент написания данной статьи он оставался однопроцессным; многопроцессные возможности были на начальной стадии реализации и, как показал наш практический опыт, их нельзя было считать работоспособными. В то же время развитие GDB в интересующем нас направлении активно обсуждается в литературе.

В [9] предложена реализация так называемой безостановочной многопоточной отладки. Безостановочность заключается в возможности отладки отдельного потока в процессе в операционной среде Linux, без остановки процесса целиком.

Такая функциональность достигается за счет расширения возможностей агента отладки, а также существенной переработки внутреннего устройства отладчика, создания механизма асинхронного обслуживания событий и контроля применимости отладочных действий. Это шаг очень важен и для развития многопроцессной отладки.

Интерактивной отладке присуща также такая проблема, как внесение недопустимо серьезных возмущений в процесс функционирования отлаживаемых систем. Этот процесс и без отладчика, вообще говоря, является недетерминированным, поэтому заранее нельзя сказать, проявятся ли ошибки в данном конкретном запуске; отладочные действия усугубляют недетерминизм, что способно привести к исчезновению предмета отладки.

Проблеме минимизации возмущений, вносимых отладочными инструментами категории GDB, посвящена статья [10]. В ней описываются различные механизмы, необходимые для многопроцессной отладки. Это централизованный ввод-вывод отладочных сообщений, параллельный сбор данных во время выполнения, использование встроенных в приложение отладочных средств и перехват передачи сообщений между процессами.

Командный язык отладки. При отладке многопроцессных систем возрастает роль командного языка отладчика. Условные точки прерывания и реакция на их достижение могут и должны программироваться заранее. Весьма вероятно, что программы отладки должны затрагивать не только непосредственно отлаживаемый процесс, но и другие процессы, функционирующие, быть может, на других процессорах. Не обязательно видеть состояние сразу всех процессов (для больших систем это не имеет смысла), но необходимо иметь возможность воздействовать на все процессы — по одиночке

или группами. Подобное групповое воздействие — обязательная черта многопроцессного отладчика.

В настоящее время в качестве командного языка отладчика применяется интерпретируемый полноценный язык программирования, такой как Python. Поддержка Python появилась в GDB версии 7.0. Наличие полноценного языка с большим числом библиотек позволяет расширить возможности отображения данных, в том числе в графическом формате, организовать взаимодействие с базами данных и реализовать сложные механизмы управления приложением.

Трассировка. Вероятно, основным инструментом отладки многопроцессных систем можно назвать трассировку. Она может иметь различную степень детализации — от машинной инструкции до программных транзакций. При аппаратной поддержке возмущения, вызываемые трассировкой, могут быть приемлемыми, а центр тяжести переносится на последующий анализ собранной информации.

При трассировке с высокой степенью детализации объем собранной информации может исчисляться гигабайтами. Для его уменьшения существуют средства сжатия трасс и, что более существенно, — условные точки трассировки, благодаря которым можно накапливать только информацию, вероятно, имеющую отношение к ошибкам и их проявлениям. В таких точках может также активизироваться интерактивный отладчик, который в данном случае остается средством "ближнего боя".

Помещать события в трассу можно не только внешними средствами, но и из программы, используя механизмы самоконтроля и генерации событий. Подобное оснащение программ представляется наиболее целесообразным, так как оно учитывает семантику программ и предполагаемую первопричину разыскиваемых ошибок, минимизирует возмущения, вносимые в процесс функционирования систем, и объем собираемой информации. И здесь возможна активизация интерактивного отладчика.

Практичный инструментальный трассировки — популярный предмет публикаций и реализаций (см., например, MkTrace [11, 12]).

Самоотладка. Для самоотладки можно предложить следующий подход. Окна интерактивной отладки ассоциируются с некоторыми "представительными" элементами аппаратно-программной конфигурации, но за счет использования групповых операций (глобальных команд `gdb`) `gstop`, `gcontinue`, `gdetach`, `greset`, `gbreak` отладка сильносвязанных систем оказывается реальной.

Многопроцессорные конфигурации, хотя они и создают сложности при отладке, могут в то же время использоваться и как инструмент отладки. При этом на части процессоров функционирует отлаживаемая система, а на других — средства отладки. Поскольку память в сильносвязанных системах явля-

ется, как правило, разделяемой, то возмущения, вносимые отладочными действиями, могут быть минимальными.

Детерминированное воспроизведение выполнения. Запуски, в которых проявляются ошибки, могут не только детерминировано воспроизводиться, но и синтезироваться как результат анализа программ. Вообще, предварительный и "посмертный" анализ программ и данных, собранных в процессе функционирования, позволяет минимизировать вносимые отладкой возмущения.

Вариантом детерминированного воспроизведения может считаться механизм создания контрольных точек останова. Такие контрольные точки могут использоваться для более быстрого воспроизведения ошибки при ее обнаружении. При этом выполнение может быть неполным, но способствовать более быстрому воспроизведению аварийной ситуации.

Интеграция средств аппаратной и программной отладки в единый комплекс контролируемого выполнения

Авторами статьи были разработаны средства аппаратной и программной отладки сильносвязанных систем и их интеграция в единый комплекс контролируемого выполнения.

В состав комплекса вошли компонент аппаратной отладки через инженерный порт, компонент интерактивной отладки, компонент сбора и анализа трассы, компонент самоконтроля и компонент воспроизведения выполнения. Все эти компоненты имеют возможность взаимодействия друг с другом при возникновении определенного события в целевой системе.

Например, компонент самоконтроля в процессе функционирования целевой системы может проверять заданные разработчиком условия на корректность, в частности, контролировать последовательность доступа к общей памяти разных процессоров. Если условия были нарушены, то компонент самоконтроля порождает определенное событие, которое перехватывается и обрабатывается компонентом сбора и анализа трассы. В свою очередь, компонент сбора и анализа трассы может инициировать сеанс интерактивной отладки или аппаратной отладки через инженерный порт. При необходимости, выполнение системы, вызвавшее сбой, может быть воспроизведено с помощью компонента воспроизведения выполнения по событиям, собранным компонентом сбора и анализа трассы.

Заключение

Набор отладочных инструментов, применяемых в отладке сильносвязанных систем, значительно шире набора инструментов традиционных последо-

вательных отладчиков. Это усложнение — необходимость, определяемая сложностью отлаживаемой системы. Без средств визуализации, трассировки, воспроизведения, самоконтроля найти ошибку зачастую невозможно. Эффективная отладка требует применения всего набора инструментов, причем каждый из них должен быть разработан с учетом сложности системы, с использованием вышечисленных подходов.

Аппаратная отладка открывает перед разработчиками новые возможности, является необходимым средством разработки приложений для ответственных сильносвязанных комплексов. Стандартизация отладки увеличивает эффективность отладочных средств, помогает их распространению. Таким образом, представляется целесообразным развивать и применять архитектуры аппаратной отладки, основанные на общедоступных спецификациях.

Важно, что подход к многопроцессной отладке требует не только применения различных инструментов, но и их интеграции в отлаживаемую систему, взаимодействия средств отладки с вычислительной средой. Эту новую по сравнению с традиционной отладкой особенность необходимо принимать во внимание не только при разработке отладчика, но и при проектировании приложений.

Список литературы

1. **Timmerman M., Gielen F., Lambrix P.** High level tools for the debugging of real-time multi-processor systems // ACM SIGPLAN Notices, 1993. Vol. 28, Is. 12. P. 151—157.
2. **MIPS EJTAG 5.00.** URL: <http://www.t-es-t.hu/download/mips/md00047f.pdf>
3. **IEEE STD. 1149.1—2001** (IEEE Standard Test Access Port and Boundary Scan Architecture). URL: <http://ieeexplore.ieee.org/iel5/481/20326/00938734.pdf>
4. **MIPS PDtrace specification.** URL: <http://www.t-es-t.hu/download/mips/md00439g.pdf>
5. **Ahmad S., Hsien-Hsin L. S.** Total recall: a debugging framework for GPUs // GH '08: Proceedings of the 23rd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware. 2008. P. 13—20.
6. **Chi-Neng W., Shu-Hsuan C., Tien-Fu C., Peisheng S. A.** NUDA: a non-uniform debugging architecture and non-intrusive race detection for many-core // DAC '09: Proceedings of the 46th Annual Design Automation Conference. 2009. P. 148—153.
7. **Mayer A., Siebert H., McDonald-Maier K. D.** Boosting Debugging Support for Complex Systems on Chip // Computer. 2007. Vol. 40, N. 4. P. 76—81.
8. **GNU Debugger.** URL: <http://www.gnu.org/software/gdb/>
9. **Non-stop Multi-Threaded Debugging in GDB.** URL: http://www.codesourcery.com/publications/non_stop_multi_threaded_debugging_in_gdb.pdf
10. **Beynon M. D., Andrade H., Saltz J.** Low-Cost Non-Intrusive Debugging Strategies for Distributed Parallel Programs // IEEE International Conference on Cluster Computing. 2002. P. 439.
11. **Li Y., Wang F., Wang G., Liu X., Liu J.** MKtrace: An Innovative Debugging Tool for Multi-Threaded Programs on Multi-processor Systems // Software Engineering Conference. APSEC 2007. 2007. P. 510—517.
12. **Moore L. J., Moya A. R.** Non-intrusive debug technique for embedded programming // Software Reliability Engineering. ISSRE 2003. 2003. P. 375—380.

V. B. Betelin, Director, e-mail: betelin@niisi.msk.ru,
V. A. Galatenko, Head of Sector, e-mail: galat@niisi.msk.ru,
K. A. Kostyukhin, Senior Researcher, e-mail: kost@niisi.msk.ru
Research Institute of System Analysis of RAS, Moscow

Controlled Execution of Tightly-Coupled Multiprocessor Systems

Development of complex software and hardware systems is a long laborious process. It is believed that about a half of the development time is spent on debugging. It makes a debugging of tightly-coupled multiprocessor systems more important and more complex.

This article discusses the controlled execution concept proposed by the authors and its application in debugging of tightly-coupled multiprocessor systems.

The main problem in debugging of tightly-coupled systems is in usage of complex asynchronous interactions between its components. This feature offers various approaches to tightly-coupled systems debugging process and determines the choice of tools and methods for debugging. The complexity of debugging is also increases due to a big number of tightly-coupled system components, software and hardware. The traditional debugging tools such as an interactive debugger, a tracer, self-control and playback libraries, and performance profiling does not lose its relevance, but its application is changed.

Keywords: tightly-coupled multiprocessor systems, controlled execution, debugging, tracing, monitoring, EJTAG

References

1. Timmerman M., Gielen F., Lambrix P. High level tools for the debugging of real-time multiprocessor systems. *ACM SIGPLAN Notices*, 1993, vol. 28, is. 12, pp. 151–157.
2. MIPS EJTAG 5.00. URL: <http://www.t-es-t.hu/download/mips/md00047f.pdf> (date: 08/04/2015).
3. IEEE STD. 1149.1–2001 (IEEE Standard Test Access Port and Boundary Scan Architecture). URL: <http://ieeexplore.ieee.org/iel5/7481/20326/00938734.pdf> (date: 08/04/2015).
4. MIPS PDItrace specification. URL: <http://www.t-es-t.hu/download/mips/md00439g.pdf> (date: 08/04/2015).
5. Ahmad S., Hsien-Hsin L. S. Total recall: a debugging framework for GPUs. *GH '08: Proceedings of the 23rd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, 2008, pp. 13–20.
6. Chi-Neng W., Shu-Hsuan C., Tien-Fu C., Peisheng S. A. NUDA: a non-uniform debugging architecture and non-intrusive race detection for many-core. *DAC'09: Proceedings of the 46th Annual Design Automation Conference*, 2009, pp. 148–153.
7. Mayer A., Siebert H., McDonald-Maier K. D. Boosting Debugging Support for Complex Systems on Chip. *Computer*, 2007, vol. 40, no. 4, pp. 76–81.
8. GNU Debugger. URL: <http://www.gnu.org/software/gdb/> (date: 08/04/2015).
9. Non-stop Multi-Threaded Debugging in GDB. URL: http://www.codesourcery.com/publications/non_stop_multi_threaded_debugging_in_gdb.pdf (date: 08/04/2015).
10. Beynon M. D., Andrade H., Saltz J. Low-Cost Non-Intrusive Debugging Strategies for Distributed Parallel Programs. *IEEE International Conference on Cluster Computing*, 2002, P. 439.
11. Li Y., Wang F., Wang G., Liu X., Liu J. MKtrace: An Innovative Debugging Tool for Multi-Threaded Programs on Multiprocessor Systems. *Software Engineering Conference. APSEC 2007*, 2007, pp. 510–517.
12. Moore L. J., Moya A. R. Non-intrusive debug technique for embedded programming. *Software Reliability Engineering. ISSRE 2003*, 2003, pp. 375–380.

УДК 004.384, 004.31

С. Г. Бобков^{1, 2}, д-р техн. наук, зам директора, e-mail: bobkov@cs.niisi.ras.ru,
О. В. Сердин, канд. техн. наук, зам. зав. отделения, e-mail: serdin@cs.niisi.ras.ru

¹ НИИ системных исследований РАН,

² НИЯУ МИФИ

Проблемы унификации микросхем космического применения на примере международного проекта космического телескопа "ГАММА-400"

Рассмотрены аспекты унификации электронных компонентов различных уровней: микросхем, интерфейсов, модулей, программного обеспечения, применяемых в космическом приборостроении. Представлена система сбора научной информации, разрабатываемая НИИСИ РАН для проекта космического телескопа "ГАММА-400". Даны основные характеристики комплекта высоконадежных и сбоеустойчивых микросхем разработки НИИСИ РАН, которые позволяют проектировать различные электронные системы космической аппаратуры.

Ключевые слова: "ГАММА-400", система сбора научной информации, КНИ-технология, система на кристалле, 8-портовый коммутатор SpaceWire, 6-портовый коммутатор Serial RapidIO, интерфейс SpaceWire, интерфейс Serial RapidIO.

Введение

Каждый спутник для научных исследований уникален и во многом из-за этого приходится под каждый эксперимент создавать уникальную вычислительную систему. Специфика области не позволяет использования коммерческих микросхем для создания вычислительных систем, серийность требуемых микросхем небольшая, а сами микросхемы должны функционировать в течение многих лет под воздействием радиации. Все это создает дополнительные проблемы, касающиеся унификации элементной базы и самих приборов. Раньше мощными толчками к стандартизации электронных изделий являлись усложнение и увеличение числа проводимых научных исследований. Достаточно вспомнить создание стандартов NIM и CAMAC. Их введение было продиктовано прежде всего ростом числа и сложности многоканальных систем сбора и накопления информации в экспериментах физики высоких энергий. В настоящее время также наблюдается значительный рост запуска научных спутников, многократно возрастает сложность экспериментов, не за горами межпланетные перелеты. В то же время стоимость научной аппаратуры значительно возрастает, очень многие эксперименты в космосе в силу их высокой сложности и стоимости становятся международными. С нашей точки зрения, все это должно привести к стандартизации как электронной аппаратуры космического применения, так и ее элементной базы. Таким образом, актуальность унификации элементной базы не вызывает сомнений.

Унификация элементной базы может происходить на уровне ядер микропроцессоров, внешних интерфейсов, коммуникационной среды, ОЗУ и внешних запоминающих устройств, схем передачи данных на Землю. Электронные модули могут унифицироваться по типоразмерам, электрическим параметрам, внешней передающей среде для связи отдельных модулей. Унификация также должна проводиться на уровне системного программного обеспечения. К сожалению, практически в каждом российском спутнике компаниями используются свои технические решения. Возьмем, например, печально известный проект космической экспедиции "Фобос-Грунт". Разработка систем управления, съема и обработки информации и передачи ее на Землю велись различными коллективами для каждого прибора. Не было единой вычислительной платформы, единых решений, обеспечивающих максимальную надежность изделий. Повсеместно использовались коммерческие компоненты иностранного производства, такие как ADSP-21060, BU-61580, AT28010, WF4M32-100G2TC5, микросхемы памяти. По официальной версии использование нерадиационно-стойких импортных микросхем памяти привело к сбою электроники в аппарате "Фобос-Грунт". Соответственно решение проблем

функционирования микропроцессоров и микросхем памяти в условиях космического пространства являются наиболее важными для успешного выполнения миссии.

В статье рассматриваются решения, принятые в НИИСИ РАН для унификации вычислительных систем космического применения.

Основные требования к элементной базе космического применения

Применение микроэлектронных изделий в условиях космического пространства предъявляет к ним ряд специальных требований, таких как: долговечность — не менее 200 тысяч часов, стойкость к накопленной дозе не ниже 200 крэд, стойкость к тяжелым заряженным частицам (ТЗЧ) с линейной передачей энергии (ЛПЭ) не ниже $100 \text{ МэВ} \cdot \text{см}^2/\text{мг}$, сохранение работоспособности после 50 нс импульсного облучения до 10^{12} рад/с, стойкость к нейтринному потоку до 10^{13} нейтрино/см², минимальная чувствительность к сбоям, вызываемым космическими частицами с высокими энергиями, т. е. речь должна вестись о создании специальной категории микросхем для космических применений. Целесообразность разработки такой категории изделий подтверждена практическим опытом ряда фирм, таких как Aeroflex, Atmel и др. Так, для построения бортовых вычислителей наиболее широко используются специальные микропроцессоры, выпускаемые компанией Aeroflex (66 МГц, технология 0,25 мкм, 5 млн транзисторов) и ф. Atmel (100 МГц, технология 0,18 мкм, 8 млн транзисторов). Оба процессора имеют уровень стойкости к накопленной дозе 300 крэд, порог по сбоеустойчивости 10...15 МэВ и уровень стойкости к эффекту зашелки $100 \text{ МэВ}/\text{мг} \cdot \text{см}$.

Для повышения устойчивости микропроцессоров к эффектам одиночных сбоев использованы специальные схемные решения функциональных элементов, а для обеспечения защиты вычислительного процесса от ошибок в процессорах применено мажорирование. В ближайшей перспективе (3...5 лет) указанные фирмы планируют начать выпуск микропроцессоров с рабочей частотой 150...200 МГц и числом транзисторов, превышающим 10 млн. Таким образом, очевидно, что обеспечение паритета России в использовании космического пространства возможно только при использовании элементной базы такого же уровня.

Дополнительными требованиями могут являться требования к широкому температурному диапазону, которые могут заметно различаться в зависимости от мест расположения аппаратуры. В настоящее время в мире существуют четыре общепринятых диапазона температур функционирования микросхем: коммерческий — 0...+70 °С, промышленный — -40...+85 °С, авиационный — -40...+125 °С, военный (в рамках стандарта MIL-STD-883) и косми-

ческий (Spice) — -55 (-60° для России) ... $+125^\circ\text{C}$, выделяются также высокотемпературные микросхемы, функционирующие при температурах свыше 150°C . Часть сложных физических детекторов для исследования космоса не может функционировать в широком диапазоне температур, соответственно требования к аппаратуре, устанавливаемой вблизи таких физических установок, аналогичны. Однако в большинстве случаев требования к комплектующим соответствуют понятию Military и Spice. Прежде всего это требования к электронике аппаратов, функционирующих на высоких орбитах и в дальнем космосе.

Одна из наиболее сложных задач — создание электронной аппаратуры для контроля и управления двигателем для полета в дальний космос. В новых подобных проектах вместо центрального компьютера используется множество распределенных микроконтроллеров и микропроцессоров. Это позволяет значительно уменьшить число и длину медных кабелей, снизить вес аппарата и увеличить надежность управления, приблизив микроконтроллер к исполнительному механизму. Кроме того, замена центрального компьютера множеством распределенных микроконтроллеров позволяет уменьшить число разъемных соединений и повысить надежность. Однако такое решение (приближение к двигателям) приводит к тому, что диапазон температур, в котором должна работать аппаратура, составляет от -60 до $+200^\circ\text{C}$. Это уже область высокотемпературной электроники. Таким образом, для таких применений должна создаваться радиационно стойкая высокотемпературная элементная база.

Унификация элементной базы и интерфейсов вычислительных систем

НИИСИ РАН свыше 10 лет занимается созданием элементной базы ЭВМ космического применения, это одно из основных направлений научной деятельности института. Проблемы унификации и стандартизации всегда были основополагающими в разработках института. Унификация проводилась по всем указанным выше направлениям. В качестве процессорных ядер использовались созданные в институте ядра с MIPS-подобной архитектурой КОМДИВ32 и КОМДИВ64. Была достигнута унификация ядер как для высокопроизводительных систем, так и для систем космического применения [1—3]. Унификация проведена также на уровне модулей [4]. В качестве базовых интерфейсов космических систем выбраны следующие:

- Space Wire для создания многопроцессорных высоконадежных вычислительных комплексов; этот интерфейс является стандартным для космических систем средней производительности;
- последовательные каналы Serial RapidIO для создания высокопроизводительных многопроцессорных систем; этот стандарт также выбран

ведущими космическими компаниями США для создания перспективных вычислительных космических систем;

- Ethernet 10/100 Мбит/с для передачи некритичных данных и совместимости с имеющимися системами;
- мультиплексный канал по ГОСТ Р 52070—2003 для обеспечения высоконадежных низкоскоростных обменов;
- RS232/RS485 для обеспечения низкоскоростных обменов, прежде всего с выпускаемыми изделиями;
- PCI для расширения функциональных возможностей модулей и организации обменов;
- JTAG для организации тестирования;
- разовые команды;
- контроллер последовательного интерфейса SPI для подключения медленных устройств.

Для создания вычислительных систем предлагается использовать два типа стандартов: VPX (конструктив Евромеханика 6U) для создания высокопроизводительных вычислительных систем и PCI/104-Express или PCIE/104 для малогабаритных систем. Стандарты позволяют строить многопроцессорные вычислительные системы различной производительности и с различными размерами и позволяют обеспечить потребности космической отрасли на 10...20 лет. Эти стандарты широко используются при создании систем промышленного, военного и космического применений в США и Европе, что дает возможность использовать модули различных производителей для верификации, тестирования и отладки программного обеспечения с последующей их заменой на отечественные аналоги.

В качестве операционной системы предлагается использование ОС РВ Багет 3.0 [5, 6]. Операционная система ОС РВ Багет 3.0 предназначена как для разработки программного обеспечения для одно- и многопроцессорных вычислительных комплексов, работающих в режиме реального времени, так и для обеспечения их функционирования.

ОС РВ Багет 3.0 представляет собой информационно безопасную и технологически независимую отечественную разработку. Эти качества подтверждаются длительным опытом эксплуатации (более 10 лет с учетом эксплуатации ОС РВ Багет 2.0) и поддержкой оборудования, разработанного в НИИСИ РАН.

ОС РВ Багет 3.0 базируется на следующих основных принципах:

- использование стандартов (спецификация ARINC 653 и стандарт POSIX);
- мобильность (возможность работы на различных аппаратных платформах);
- гибкие средства планирования, включающие как периодические вычисления, так и использование приоритетов;

- развитые средства диагностики и обработки ошибок, а также восстановления работоспособности после сбоев (монитор здоровья);
- управляемость (в частности, наличие средств конфигурирования).

При разработке ОС РВ использовалась спецификация ARINC 653 и стандарт POSIX 1003.1. Как POSIX, так и ARINC 653 определяют интерфейс прикладных программ с операционной системой. Этим обеспечивается мобильность создаваемых приложений.

В целях повышения мобильности операционной системы она разбита на три части:

- часть ОС РВ, не зависящая от аппаратуры;
- часть ОС РВ, зависящая только от типа центрального процессора;
- пакет поддержки модуля.

Первая и большая часть ОС РВ написана на языке Си и не зависит от аппаратной части вычислительного комплекса.

Вторая часть ОС РВ, которая зависит только от типа процессора, написана на языке Си или на Ассемблере и имеет сравнительно небольшой объем.

Пакет поддержки модуля (ППМ) содержит ту часть ОС, которая зависит от конкретной ЭВМ (процессорного модуля). ППМ, в частности, содержит драйверы устройств и частично диспетчер прерываний.

Система сбора и хранения научной информации проекта "ГАММА-400"

Применение комплексных решений унификации осуществлено в разрабатываемой системе сбора и хранения научной информации комплекса научной аппаратуры "ГАММА-400" (ССНИ). Научный комплекс "ГАММА-400" [7] предназначен для получения данных для определения природы "темной материи" во Вселенной, развития теории происхождения высокоэнергичных космических лучей и физики элементарных частиц, исследования космического гамма-излучения в диапазоне высоких энергий, регистрации заряженных частиц космических лучей, поиска и исследования гамма-всплесков.

Основными функциями ССНИ являются:

- прием информации от научных подсистем комплекса научной аппаратуры (КНА);
- предварительная обработка научной информации и сохранение ее в энергонезависимой массовой памяти;
- передача информации в радиоканал космического аппарата (КА) во время сеансов связи с Землей;
- прием информации от бортового командного устройства и передача ее в научные приборы и их подсистемы.

Ежедневно планируется передача информации на Землю в течение сеанса связи в объеме 100 Гбайт. Для обеспечения устойчивой передачи в течение

всего срока службы (7...10 лет) планируется создание накопителя объемом 1 Тбайт. Огромные объемы передаваемой информации приводят к требованиям создания высокопроизводительных вычислительных систем с высокоскоростными каналами обмена данными. ССНИ является ключевым узлом (КНА) "ГАММА-400" и, следовательно, должна быть высоконадежным элементом. В разрабатываемой вычислительной системе используются как каналы SpaceWire со скоростью передачи до 200 Кбит/с, так и Serial RapidIO 4X со скоростью передачи 1,25 Гбит/с на линию (1 Гбайт/с на канал).

Основной тенденцией в создании систем КНА является унификация и стандартизация оборудования, в том числе и интерфейсов передачи данных. В разрабатываемой системе предполагается большое число каналов (до 16) от разнородных источников цифровых массивов. Прием данных должен осуществляться по надежному высокоскоростному каналу, которым является интерфейс SpaceWire. Наличие коммутаторов каналов SpaceWire позволяет масштабировать вычислительную систему.

В проекте "ГАММА-400" выделен объем порядка $1 \times 0,5 \times 0,5$ м под создаваемую многопроцессорную вычислительную систему, что позволяет использовать стандарт VPX с размерами печатных плат 160×233 мм (Евромеханика 6U).

В целях обеспечения требуемой надежности в ССНИ применяется "перекрестное" резервирование, т. е. основной и резервный полукомплекты ССНИ могут использовать подключения к дублированной высокоскоростной линии (ВРЛ) и к основному, и резервному долговременному запоминающему устройству (ДЗУ), каждое объемом по 500 Гбайт.

Схема одного полукомплекта ССНИ представлена на рис. 1. Сбор данных с научных детекторов (на рис. 1 обозначены как ИЦМ — источник цифровых массивов) осуществляется по 16 высокоскоростным каналам SpaceWire (на рис. обозначены SW1-16). Общая требуемая пропускная способность каналов сбора научной информации (НИ) составляет 70 Мбайт/с. НИ через устройство коммутации (Коммутатор SW) передается по каналам

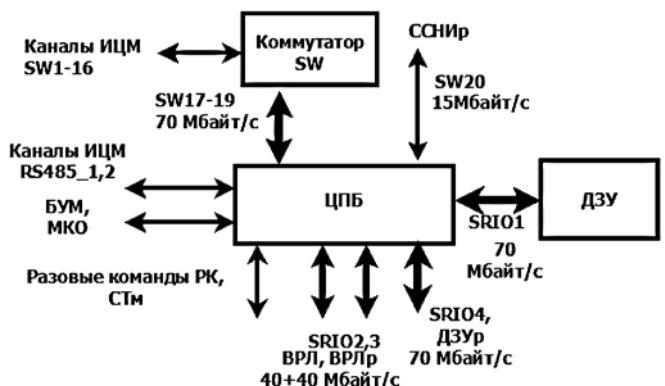


Рис. 1. Схема полукомплекта ССНИ

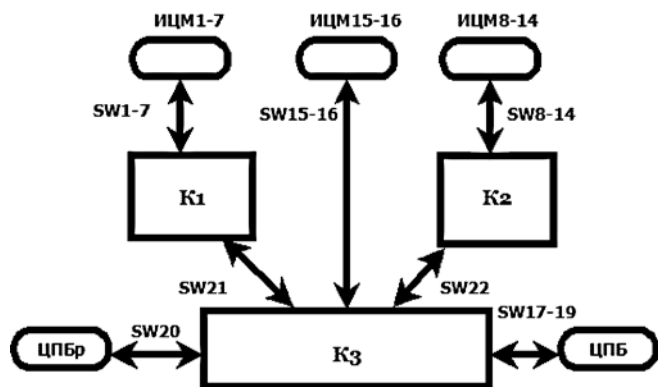


Рис. 2. Схема модуля коммутатора SW

SpaceWire (SW17-19) для обработки в центральный процессорный блок (ЦПБ), в состав которого входит управляющий микропроцессор на базе микросхемы 1907BM038. Микропроцессор 1907BM038 через 6-портовый коммутатор Serial RapidIO (микросхема 1907KX018) одновременно обеспечивает сохранение обработанной НИ в ДЗУ (канал SRIO1) и выборку ранее сохраненных данных из ДЗУ для передачи их на наземный комплекс управления через ВРЛ (канал SRIO2). ДЗУ выполнено на микросборках из микросхем NAND-флэш фирмы 3Dplus. Высокий уровень надежности ССНИ достигается минимальным необходимым числом высокоинтег-

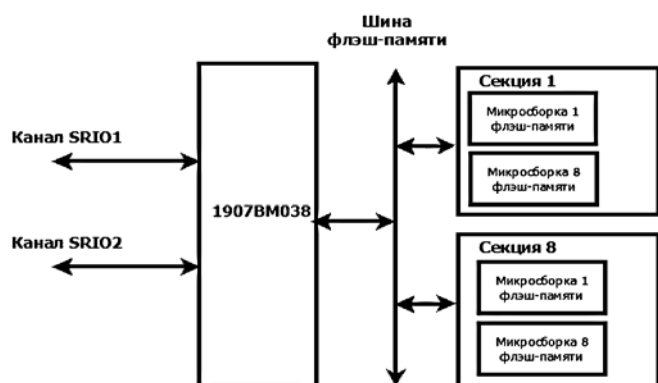


Рис. 3. Схема ДЗУ

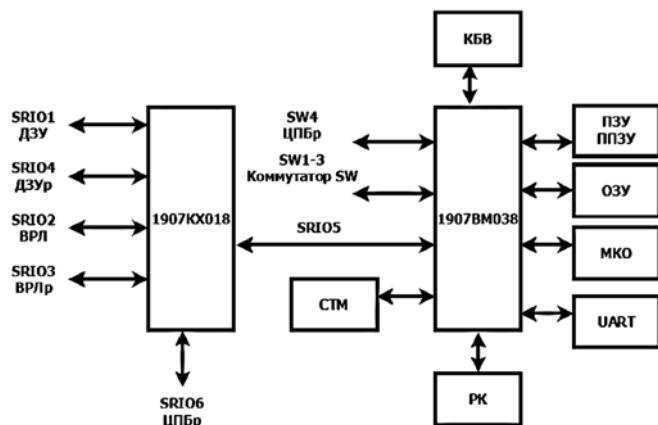


Рис. 4. Схема ЦПБ

рированных микросхем, разрабатываемых коллективом НИИСИ РАН.

Связь с бортовым комплексом управления (БКУ) ССНИ осуществляет через резервированный мультиплексный канал обмена (МКО) и дублированные разовые команды. Система телеметрии (СТМ) обеспечивает проверки исправности отдельных узлов, контроль температур наиболее теплонагруженных микросхем, доставку результатов телеметрического контроля от научных приборов.

На рис. 2 представлена схема модуля коммутатора. Основой модуля являются три микросхемы восьмиканального коммутатора SpaceWire (K₁—K₃), проектируемого в НИИСИ РАН по технологическим нормам КМОП КНИ 0,25 мкм.

Схема ДЗУ представлена на рис. 3. Все необходимые функциональные блоки для управления массивом флэш-памяти интегрированы в микросхему 1907BM038.

На рис. 4 представлена схема модуля ЦПБ. Управляющий микропроцессор 1907BM038 способен выполнять все необходимые системные функции:

- прием и проверка данных по каналам SpaceWire от коммутатора SW;
- сортировка, контроль расхода выделенных информационных объемов и отправка данных по каналам Serial RapidIO либо в ДЗУ, либо в радиолинию (ВРЛ);
- прием от БКУ бортовой шкалы времени (БШВ) и генерация кода бортового времени (КБВ) и меток времени для научных приборов;
- прием от БКУ управляющих кодов по мультиплексному каналу (МКО) и трансляция кодов по сети SpaceWire научным приборам;
- прием от БКУ управляющих разовых команд (РК) и обеспечение телеметрической информации (СТМ), поступающей на БКУ;
- периодический контроль исправности основных микросхем, а также выполнение самодиагностики;
- управление резервированием отдельных блоков, в том числе подключение обходных путей передачи данных в сетях SpaceWire и Serial RapidIO.

Сеть Serial RapidIO основана на коммутаторе 1907KX018. Коммутатор последовательных каналов в неблокирующем режиме способен коммутировать одновременные потоки данных от управляющего микропроцессора в режиме записи данных в ДЗУ и чтения данных из ДЗУ и отправки их в ВРЛ. Число портов коммутатора достаточно для подключения резервных блоков радиолинии (ВРЛр) и массовой памяти (ДЗУр).

Основные характеристики базовых микросхем проекта "ГАММА-400"

Система на кристалле 1907BM038 является перспективной микросхемой для создания различных систем бортового космического применения, срок

окончания ОКР — 2015 г. Технология изготовления — КНИ 0,25 мкм, завод-изготовитель — 1X1 (НИИСИ РАН). Структурная схема СнК представлена на рис. 5.

В состав системы входят следующие основные блоки:

- 32-разрядное управляющие ядро (CPU);
- 128-разрядный арифметический сопроцессор (на рис. обозначен как CP2);
- системный контроллер с набором интерфейсов: DDR2, SPI, два канала Serial RapidIO, четыре канала SpaceWire, I2C, GPIO, два контроллера UART.

Тактовая частота микросхемы — 100 МГц. Пропускная способность ОЗУ — 512 Мбайт/с.

Конструкция корпуса микросхемы: металлокерамический корпус типа dumdled BGA с числом выводов — 675 шт.

Микросхема коммутатора последовательных каналов 1907KX018 предназначена для организации связи между устройствами (коммутаторами и системы на кристалле, СнК), имеющими в своем составе каналы последовательного RapidIO. Коммутатор спроектирован в НИИСИ РАН, является перспективной микросхемой для создания высокопроизводительных систем бортового космического применения. Технология изготовления — КНИ 0,25 мкм, завод-изготовитель — 1X1 (НИИСИ РАН).

Коммутатор содержит шесть портов последовательного RapidIO, среда передачи: LP-Serial 4X либо 1X. Максимальная скорость передачи по каждому каналу — 1,25 Гбит/с. Структурная схема микросхемы представлена на рис. 6. Отличительными характеристиками являются: отдельная таблица коммутации для каждого порта, система контроля производительности, встроенный блок управления ошибками. Коммутатор способен непосредственно соединять до 256 оконечных устройств (СнК, контроллеры ввода-вывода и т. п.) в системе, что обычно является достаточным для бортовых систем. Индивидуальные таблицы коммутации позволяют гибко настраивать пути прохождения пакетов по сети. Система контроля производительности используется для определения характеристик потока данных в канале, определения перегрузок в сети, определения мест блокировок. Определение величины потока данных в канале проводится при выборе топологии сети на этапе моделирования и проведении натурного эксперимента. Причинами перегрузок в сети могут быть ухудшение состояния канала, возникшее в результате сбоев. Поэтому раннее оповещение об ошибках уменьшает время, необходимое на подключение резервного пути. Для выбора обходного пути крайне полезной является

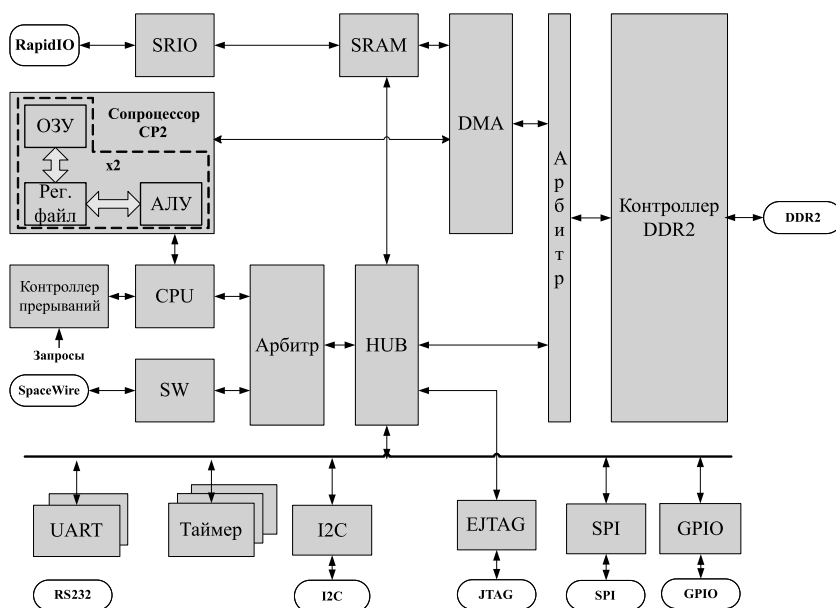


Рис. 5. Структурная схема микросхемы микропроцессора 1907BM038

способность коммутатора определять заблокированные каналы.

Максимальная потребляемая мощность при использовании всех шести портов в режиме 4X не более 6 Вт. Предусмотрено снижение потребляемой мощности в режимах работы портов 1X, а также путем индивидуального отключения передатчиков каждого порта. Микросхема имеет два канала питания: 2,5 В и 3,3 В.

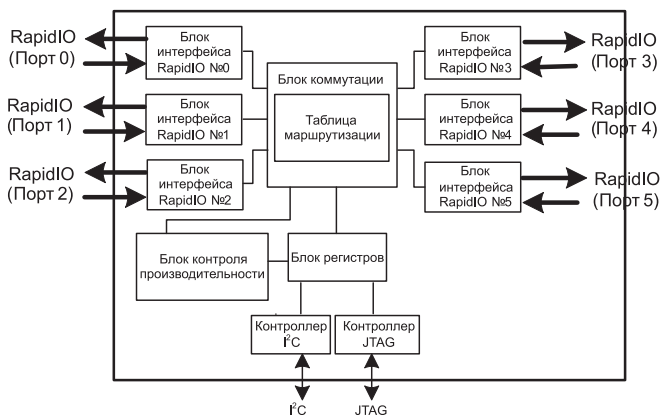


Рис. 6. Структурная схема и фотография микросхемы коммутатора RapidIO 1907KX018

Конструкция корпуса микросхемы: керамический корпус типа dumdled BGA с числом выводов 399 шт.

В создаваемой системе число необходимых внешних компонентов сведено к минимуму, что позволяет соответственно поднять надежность системы. Таблицы коммутации можно либо запрограммировать через порты последовательного RapidIO, либо загружать из последовательного EEPROM через интерфейс I2C.

Повышенная стойкость к сбоям достигается комплексом дополнительных мер:

- помехоустойчивое кодирование при обращении к памяти буферов пакетов;
- использование резервирования таблиц маршрутизации;
- топологические и схемотехнические решения библиотечных элементов.

Коммутатор имеет расширенные возможности диагностики работоспособности. Состояние диагностических регистров доступно через интерфейс I2C. Кроме того, отдельные интегрированные сигналы сбоев и ошибок могут быть выведены на порт дискретных сигналов для дальнейшего анализа системой контроля исправности (СКИ), которая в данном случае упрощается, соответственно повышается надежность.

Микросхема многопортового коммутатора SpaceWire со встроенным микропроцессором (ОКР "Схема-23") проектируется в НИИСИ РАН, срок окончания ОКР — 2016 г. Технология изготовления — КНИ 0,24 мкм, завод-изготовитель — ОАО "НИИМЭ и завод Микрон". Структурная схема коммутатора представлена на рис. 7.

Тактовая частота микросхемы — 100 МГц. Максимальная потребляемая мощность не более 6 Вт. Предусмотрены режимы отключения неиспользуемых функциональных блоков в целях понижения потребляемой мощности.

При выполнении функций коммутатора для работоспособности микросхемы требуется малое

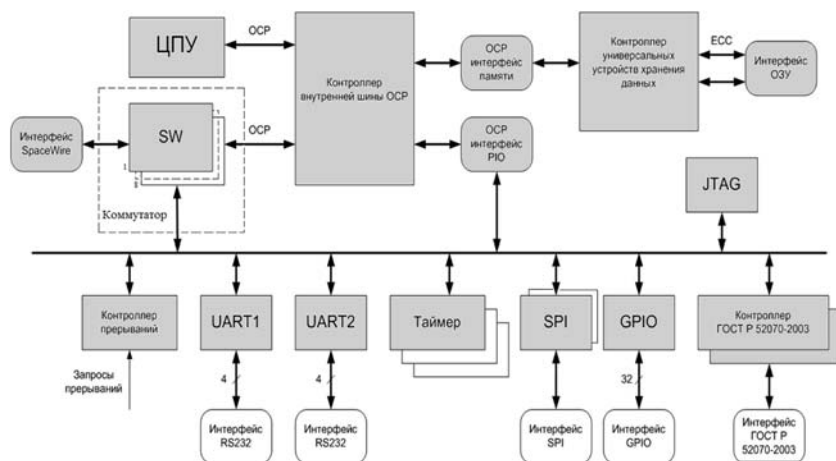


Рис. 7. Структурная схема многопортового коммутатора SpaceWire со встроенным микропроцессором

число элементов, например, не требуются микросхемы внешнего ОЗУ.

Для определения возможности применения рассмотренных микросхем в космических аппаратах был проведен комплекс исследований. Были получены следующие характеристики:

- пороговое значение линейных потерь энергии тяжелых заряженных частиц (ТЗЧ) по тиристорному эффекту не менее $80 \text{ МэВ} \cdot \text{см}^2/\text{мг}$ (эффект не наблюдался);
- пороговое значение линейных потерь энергии ТЗЧ по одиночным событиям не менее $6 \text{ МэВ} \cdot \text{см}^2/\text{мг}$;
- сечение насыщения по одиночным событиям при воздействии ТЗЧ — не более $3 \cdot 10^{-8} \text{ см}^2/\text{бит}$;
- уровень стойкости к накопленной дозе — 500 крад.

Высокие параметры по стойкости объясняются прежде всего созданием в НИИСИ РАН специального технологического процесса производства микросхем с нормами 0,25...0,35 КНИ, а также разработкой радиационно стойкой библиотеки. Диапазон температур функционирования микросхем от -60° до $+125^\circ \text{C}$. Таким образом, созданные микросхемы позволяют решить большую часть задач создания вычислительных систем космического применения и с огромным запасом покрывают потребности проекта "ГАММА-400".

Для определения возможности использования микросхем в высокотемпературных приложениях были проведены исследования микросхем при температуре функционирования 150°C . Получены следующие результаты. При данной температуре частота функционирования в среднем уменьшается на 10 %, рост потребляемой мощности не наблюдался, однако это связано, вероятно, с падением частоты функционирования. Длительность функционирования в таком режиме не определялась. Для создания полноценного процесса изготовления высокотемпературных микросхем необходима модернизация технологического процесса: изменение типа примеси, утолщение проводников и пр. Тем не менее, показана принципиальная возможность создания высокотемпературных микросхем космического применения на базе имеющихся проектов.

Заключение

Таким образом, достигнута унификация микросхем космического применения на уровне ядра, внешних интерфейсов и среды обмена данными. Для сокращения потребляемой мощности и повышения надежности среда передачи данных сделана иерархической. На верхнем уровне находится коммутируемая среда RapidIO со скоростью обмена до 1 Гбайт/с на канал. На следующем уровне — коммутируемая среда

SpaceWire со скоростью обмена до 200 Мбит/с и на нижнем уровне низкоскоростные мультиплексный канал по ГОСТ Р 52070—2003 и RS232/RS485. Интерфейс Ethernet является технологическим и предусмотрен также для связи с ранее выпускаемыми модулями. Коммутируемые среды позволяют создавать различные архитектуры системы с резервированием в рамках стандартных решений для таких сред.

Для создания вычислительных систем космического применения предложено использовать два типа стандартов: VPX (конструктив Евромеханика 6U) для создания высокопроизводительных вычислительных систем и PCI/104-Express или PCIe/104 для малогабаритных систем. В проекте "ГАММА-400" вычислительная система создается на базе стандарта VPX.

В качестве операционной системы предложено использование отечественной ОС РВ Багет 3.0. Операционная система сделана в соответствии с международными стандартами, переносима на различные платформы и апробирована в многочисленных системах (несколько сотен).

Работа выполнена в рамках проекта по созданию вычислительной системы космического телескопа "ГАММА-400" и при поддержке гранта РФФИ

№ 14-29-09207 офи_м "Фундаментальные аспекты конструктивно-схемотехнической организации элементов радиационно-стойких запоминающих устройств для высокотемпературной микроэлектроники".

Список литературы

1. **Бобков С. Г.** Архитектура микропроцессоров суперЭВМ эксафлопсного диапазона // Информационные технологии. 2012. № 12. С. 2—9.
2. **Соколов И. А., Степченко Ю. А., Бобков С. Г., Захаров В. Н., Дьяченко Ю. Г., Рождественский Ю. В., Сурков А. В.** Базис реализации супер-ЭВМ эксафлопсного класса // Информатика и ее применения, 2014. Т. 8, Вып. 1. С. 47—72.
3. **Бобков С. Г.** Импортозамещение элементной базы вычислительных систем // Вестник Российской Академии Наук. 2014. Т. 84, № 11. С. 1010—1016.
4. **Бобков С. Г., Ерёмин А. В., Катаев А. Ф., Комиссаров П. В., Кондратьева Н. В., Сердин О. В.** Разработка высоконадежных многопроцессорных модулей на базе высокоскоростных каналов RapidIO // Программные продукты и системы. 2013. № 4. С. 49—55.
5. **Годунов А. Н.** Операционная система реального времени Багет 3.0 // Программные продукты и системы. 2010. С. 16—20.
6. **Годунов А. Н., Солдатов В. А.** Операционные системы семейства Багет (сходство, отличия и перспективы) // Программирование. 2014. № 5. С. 68—76.
7. **Adriani O., Galper A. M., Bonvicini V., Topchiev N. P., Bobkov S. G.** and et. The GAMMA-400 Space Observatory: Status And Perspectives. URL: <http://arxiv.org/abs/1412.4239>, Submitted on 13 Dec. 2014.

S. G. Bobkov, Deputy Director, SRISA, e-mail: bobkov@cs.niisi.ras.ru,

O. V. Serdin, Deputy Head of Division, SRISA, e-mail: serdin@cs.niisi.ras.ru

Unification of Chip Space Application on the Example of International Project Space Observatory "GAMMA-400"

The problems of electronic component unification at the different levels (circuits, interfaces, modules, software used in space industry) are considered. The task of computer systems for space purposes developing is considered. The system of collecting scientific information (SCSI) developed by SRISA for the space telescope GAMMA-400 project is presented. The basic characteristics of the set of high reliable and fault tolerant chips developed by SRISA for the computational systems design for space purposes is given.

Achieved the chips unification for space applications for the kernel level, external interfaces and Data Acquisition System. To reduce power consumption and enhance data reliability embedded system interconnect is made hierarchical: upper level is Serial RapidIO $\times 4$ or $\times 1$ with rate transfer 1.25 Gbaud per line; next level — SpaceWire with rate transfer up to 200 Mbaud and lower level — MIL-STD-1553B and RS232/RS485. The Ethernet 10/100 is technology interface and provided connection with the previously released modules. System interconnect allows to create different redundancy system. Designers can develop heterogeneous systems that leverage the peer-to-peer networking performance of RapidIO while at the same time using multiprocessor clusters that may only be SpaceWire enabled. The first application of the system is "GAMMA-400" gamma-ray telescope, measuring the energy spectra of Galactic and extragalactic diffuse gamma-ray emission, studying gamma-ray bursts and gamma-ray emission from the Sun.

Keywords: "GAMMA-400", the system of collecting scientific information, SOI technology, system on chip, the SpaceWire switch, the Serial RapidIO switch, SpaceWire interface, Serial RapidIO interface

References

1. **Bobkov S. G.** Архитектура микропроцессоров суперЭВМ эксафлопсного диапазона. *Informacionnue tekhnologii*, 2012, no. 12, pp. 2—9.
2. **Sokolov I. A., Stepchenkov U. A., Bobkov S. G., Zakharov V. N., Diachenko U. G., Rozhdestvensky U. V.** Базис реализации суперЭВМ эксафлопсного класса. *Informatika i eje Primenenie*, 2014, vol. 8, iss. 1, pp. 47—72.
3. **Bobkov S. G.** Importozameshenie elementnoi bazu vuchislitelnych sistem. *Vestnik Rossiiskoi Akademii nauk*, 2014, vol. 84, no. 11, pp. 1010—1016.

4. **Bobkov S. G., Eremin A. V., Kataev A. F., Komissarov P. B., Kondratieva N. V., Sedin O. V.** Razrabotka vusokonadezhnykh mnogoprocessornukh modulei na baze vusokoskorostnykh kanalov RapidIO. *Programmnie produktu i sistemy*, 2013, no. 4, pp. 49—55.
5. **Godunov A. N.** Operacionnaia Sistema realnogo vremeni Baget 3.0. *Programmnie produktu i sistemy*. 2010, no. 4, pp. 16—20.
6. **Godunov A. N., Soldatov V. A.** Operacionnue sistemu semeistva Baget (shodstva, otlichia, i perspektivu). *Programmirovanie*, 2014, no. 5, pp. 68—76.
7. **Arinani O., Galper A. M., Bonvicini V., Topchiev N. P., Bobkov S. G.** The GAMMA-400 Space Observatory: Status and Perspectives. URL: <http://arxiv.org/abs/1442.4239>. Submitted on 13 Dec 2014.

М. Е. Барских¹, зав. сектором, e-mail: barskikh@cs.niisi.ras.ru,
 С. Г. Бобков^{1, 2}, д-р техн. наук, зам. директора, e-mail: bobkov@cs.niisi.ras.ru
¹ НИИ системных исследований РАН,
² НИЯУ МИФИ

Исследование влияния динамического предсказания ветвлений на производительность перспективных микропроцессоров НИИСИ РАН

Проведено исследование влияния различных способов предсказания команд условного перехода на производительность микропроцессоров с RISC-архитектурой КОМДИВ. Проведены сравнительные исследования влияния параметров схемы динамического предсказания на ее точность. Подготовлена методика проведения подобных исследований и обоснована корректность реализации блока.

Ключевые слова: суперскалярный процессор, динамическое предсказание переходов, моделирование

Введение

Основными факторами сдерживания производительности микропроцессоров являются рост потребляемой мощности и снижение надежности. Во многом данные параметры определяются архитектурой микропроцессоров. В связи с этим в настоящее время все большую популярность завоевывают микропроцессоры с RISC-архитектурой. Производительность систем, построенных на базе RISC-процессоров, практически не уступает процессорам с архитектурой x86 при существенно меньшей стоимости и потребляемой мощности (для примера в табл. 1 приведено сравнение характеристик систем с процессорами ARM и с процессорами Sandy Bridge компании Intel).

Производительность микропроцессоров с RISC-архитектурой во многом определяется возможностями параллельного выполнения нескольких инструкций. Возможность одновременного выполнения инструкций решается, в частности, механизмом предсказания переходов. Суть метода заключается в прогнозировании направления ветвления программы, запросе и начале выполнения команд по адресу перехода. Повышение эффективности данного метода является актуальным для любого современного микропроцессора. Это объясняется тем, что в обычных программах порядка 15 % [1] инструкций являются командами ветвления, которые в простейших случаях реализации архитектуры приводят к остановке конвейера. В современных микропроцессорах глубина конвейера достигает

нескольких десятков тактов [2—4], и остановка конвейера приводит к потере этих тактов и к ответственному снижению производительности. Базовые подходы к механизму предсказания ветвления одинаковые для всех RISC-микропроцессоров — это статическое и динамическое предсказание ветвления. Однако реализация этой функции для разных микропроцессоров разная и занимает в случае динамического предсказания, как правило, от двух до пяти тактов, что становится сопоставимым с самим вычислением и приводит к соответствующему повышению потребляемой мощности. Поэтому оптимизация блока предсказания ветвления для каждого микропроцессора проводится индивидуально.

Можно дать следующее определение модулю предсказания условных переходов (*branch prediction unit*, BPU) — устройство, входящее в состав микропроцессоров, имеющих конвейерную архитектуру, которое определяет направление ветвлений (предсказывает, будет ли выполнен условный переход) в исполняемой программе. Предсказание ветвлений позволяет осуществить предварительную выборку инструкций и данных из памяти и начать выполнение инструкций, находящихся после условного перехода, до того, как он будет выполнен. Предсказатель переходов является неотъемлемой частью всех современных суперскалярных микропроцессоров, так как в большинстве случаев (точность предсказания переходов в современных процессорах превышает 90 %) позволяет оптимально использовать вычислительные ресурсы процессора.

Таблица 1

Сравнительные характеристики процессоров с архитектурами RISC и Intel

Микро-процессор	Производительность, Гфлопс	Число адресных пространств	Объем накристалльной памяти, Мбайт	Пропускная способность памяти, Гбайт/с	Пропускная способность коммуникационных каналов, Гбайт/с	Стоимость, \$	Мощность, Вт
2×Sandy Bridge	370	1	44	136	56	>3000	>350
8×ARM	256	8	16	102	40	<200	<100

Существуют два основных метода предсказания переходов: статический и динамический.

Статические методы предсказания ветвлений являются наиболее простыми. Суть этих методов состоит в том, что переходы различных типов либо выполняются всегда, либо не выполняются никогда. При реализации статических предсказаний предполагается, что любой переход "назад" (т. е. переход на младшие адреса) является циклом и должен выполняться, а любой переход "вперед" (т. е. на старшие адреса) не выполняется. Динамические методы, широко используемые в современных процессорах, подразумевают анализ истории ветвлений, обновляемой по результатам фактического выполнения команд переходов.

Поэтому для применения в разрабатываемых в НИИСИ перспективных микропроцессорах с MIPS-подобной архитектурой КОМДИВ [5, 6] был разработан блок динамического предсказания ветвления. Целью данной работы была верификация этого блока, т. е. проверка, что он "работает" и "работает хорошо". В качестве критериев использовались следующие характеристики: точность предсказания переходов (команд ветвления) и производительность ядра процессора в виде среднего числа выполняющихся команд за такт (*instruction per cycle*, IPC). Проверка того, что блок динамического предсказания переходов "работает", заключалась в подтверждении того, что точность предсказания и значение IPC выше, чем в случае статического предсказания и, тем более, без него. Так как существует множество параметров, влияющих на работу блока динамического предсказания, следующей целью работы было определение лучшего их сочетания. В этом случае можно говорить о том, что блок "работает хорошо", т. е. мы получаем максимальную точность предсказаний при одинаковых аппаратных затратах на реализацию этого блока. Вместе с тем, погоня только за точностью предсказания может приводить к необоснованному усложнению блока и, соответственно, к росту потребляемой мощности. Поэтому использование в качестве критерия IPC позволяет определить момент, когда усложнение блока уже практически не влияет на рост производительности и на данной реализации можно остановиться.

Выбор схемы динамического предсказания

Выбор схемы предсказания основывался на анализе решений, применявшихся в коммерческих процессорах. К сожалению, в последнее время технических данных об используемых схемах предсказания немного. Это связано с тем, что выбор оптимальной схемы — очень кропотливое и долгое занятие, а результат является коммерческой тайной, так как позволяет увеличить производительность процессора. При этом улучшения тем выше, чем длиннее конвейер и чем большее окно выполнения он имеет. Кроме этого, в последнее время в про-

цессорах используются не просто схемы предсказаний направления перехода, а схемы, предсказывающие целевой адрес перехода и использующие буфер целевых адресов (*branch target buffer*, ВТВ). Это необходимо в случае, когда только запрос инструкции занимает несколько тактов и необходимо предсказывать не только направление, но и адрес перехода для конвейерной выборки инструкций. Однако сложность ВТВ-буфера сопоставима со сложностью кэш-памяти инструкций. Наиболее полное перечисление используемых механизмов предсказаний переходов в микропроцессорах x86 приведены в [7].

Проведенный анализ показал, что наиболее оптимальной из известных схем динамического предсказания переходов с точки зрения сложности и объема схемы, потребления питания и получаемых результатов предсказания является двухуровневая гибридная схема динамического предсказания переходов, реализованная в микропроцессорах Alpha 21264 [8, 9], использованная затем в процессорах POWER4 [10] и POWER5 [11]. Для дополнительного снижения потребления питания в предлагаемой нами схеме локального предсказания не используется память локальной истории переходов, а сам счетчик предсказания — двухразрядный. Общая логическая схема блока представлена на рис. 1.

Для обеспечения возможности анализировать различные реализации блока предсказаний схема сделана конфигурируемой. Конфигурационные регистры, доступные в том числе и программно, позволяют изменять:

- тип предсказания (динамическое, статическое и отключено);
- вид динамического предсказания;
- алгоритм обновления истории;
- размер памятей истории;
- варианты адресации (выбора данных) памятей истории.

В результате блок может быть переконфигурирован для исследования большого числа вариантов схем динамического предсказания, в частности, для проведения исследований схемы *gskew* [12], которая разрабатывалась для микропроцессора Alpha EV8 [13].

Благодаря относительно короткому конвейеру в семь стадий (из которых запрос инструкции со-

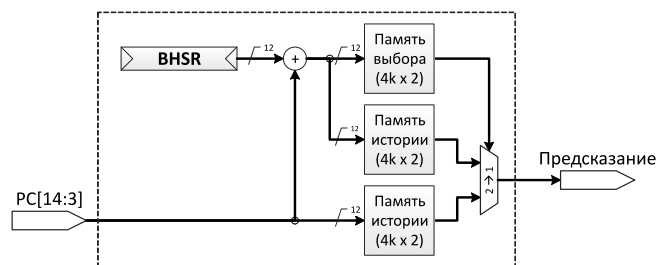


Рис. 1. Общая схема работы блока динамического предсказания (BHSR — branch history shift register, сдвиговый регистр истории переходов; PC — program counter, программный счетчик)

ставляет две стадии) в такой схеме нет необходимости использовать ВТВ. Расчет адреса перехода делается в момент получения инструкций из кэш-памяти, в следующем такте выставляется новый запрос по предсказанному (или следующему, если предсказания не было) адресу. Дополнительно с блоком динамического предсказания, для команд безусловных регистровых переходов сделан буфер возврата на восемь адресов.

Данные о точности предсказаний и производительности накапливаются в специальных регистрах, доступных для чтения.

Конвейер процессора и запрос инструкций

На рис. 2 показаны две первые стадии конвейера процессора, в которых происходит запрос инструкций из кэш-памяти. Параллельно с запросом инструкций выполняется расчет предсказания перехода и рассчитывается целевой адрес перехода. Для расчета адреса смещения берутся из кода полученных инструкций (для условных и не регистровых безусловных переходов) или из стека возврата (для регистровых безусловных переходов). Это было возможно, так как кэш-память инструкций работает на сдвинутом на половину периода тактовом сигнале.

В части запроса инструкций процессор имеет две особенности, которые необходимо было учитывать при разработке: у всех инструкций перехода MIPS-архитектуры есть слот задержки, который выполняется после этой инструкции; и в случае попадания в кэш инструкций буфер автоматически получит инструкции по текущему и "инкрементированному" запросу. Наличие слота задержки и его обязательное выполнение приводят к тому, что если он не был получен в текущем запросе, следующий запрос будет по адресу слота задержки, а запрос по предсказанному адресу будет отложен.

Возможность получения двух групп инструкций по одному запросу обуславливает необходимость контролировать момент выставления предсказан-

ного запроса для отмены записи в буфер инструкций, полученных от предыдущего запроса. Эти ситуации контролируются конечным автоматом запросов, который получает информацию от блока предсказаний. Кроме этого, автомат запросов отслеживает отмену запросов, связанных с исключительными ситуациями, или восстановление после неправильного предсказания. Однако из-за унификации поведения буфера инструкций в этих двух ситуациях и благодаря одинаковому механизму работы при предсказании любого типа ветвлений (условного, безусловного или вероятного (*branch likely*) перехода), сложность машины состояний по сравнению с версией 1890VM6 даже уменьшилась.

Так как на вычисление адреса остается половина такта, что недостаточно для полного 64-разрядного сумматора, адрес вычисляется в два этапа (такта). На первом такте вычисляются инкрементированный и декрементированный адреса 48-битной базы, общей для всех получаемых инструкций, и 16-битные смещения базового адреса для каждой из четырех получаемых инструкций. Результаты сохраняются в регистры и используются на следующем такте, когда из кэш-памяти поступают считанные инструкции. Младшие 16 бит всех инструкций суммируются с соответствующими смещениями базы (адреса запроса) и на основании знака смещения инструкции и флагов переносов двух сумматоров смещений выбирается соответствующая старшая часть базового адреса.

Стек возврата представляет собой кольцевой буфер на восемь записей для сохранения последних адресов, генерируемых командами *jump & link* (ими делается переход в подпрограммы в MIPS-архитектуре). Чтение из него происходит командами безусловных регистровых переходов, которые делают возврат из подпрограмм. От "классического" стека стек возвратов отличается тем, что он может быть считан, когда в нем нет записей, и записан, когда он полон (так как выполнение инструкций ничем не ограничено).

При записи в заполненный стек затирается самая первая запись, соответствующая самому старому адресу возврата, и работа продолжается в нормальном режиме (буфер инструкций даже не имеет информации об этом событии). При получении команды регистрового перехода целевой адрес для нее берется из стека возврата, и команда считается предсказанной. После фактического выполнения команды рассчитанный адрес перехода сравнивается с предсказанным, и в случае ошибки инструкции перезапрашиваются. В случае пустого стека дальнейшее выставление запросов блокируется до фактического выполнения команды перехода и вычисления его адреса уже на стадии выполнения.

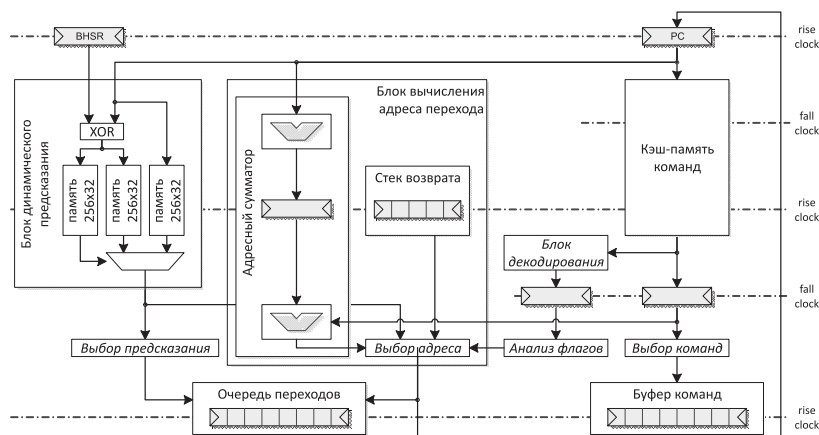


Рис. 2. Стадии запроса инструкций и предсказания перехода (первые два такта конвейера)

Для моделирования была взята RTL модель системы на кристалле (в состав которой входят системный контроллер и ядро процессора). К модели подключена память DDR2, соотношение частот при моделировании: 500 МГц для ядра процессора и 250 МГц для памяти DDR2. Моделирование проводилось на аппаратном ускорителе Palladium XP фирмы Cadence. После компиляции проект имел реальную частоту 1,7 МГц. Это позволило моделировать тесты длительностью примерно 5 млн инструкций, что достаточно как для начального накопления истории переходов, так и для оценки работы схемы предсказаний.

Принципиальная схема предсказания переходов, реализованная для предлагаемого блока, с управляющей логикой, с помощью которой происходило конфигурирование схемы, показана на рис. 3. Для моделирования были отобраны программы, выполняющие различные операции (табл. 2), структура тестов (пропорции команд в тесте по группам) представлена на рис. 4 (см. третью сторону обложки). В качестве тестов были использованы как широко известные тесты производительности (Coremark, Dhrystone и др.), так и программы, реализующие прикладные алгоритмы (gZip, MatrixMult). Выбор тестов обусловлен необходимостью иметь возможность ограничивать их длительность без ущерба для вычислительного ядра программы, т. е. чтобы функциональность программы не пострадала. В отличие от тестов Sres, они были небольшими, не требовали поддержки со стороны операционной системы (что не только замедляет моделирование, но и вносит погрешность из-за накопления истории пред-

Тесты, использовавшиеся при моделировании

Тест	Описание	Длительность теста	
		Число команд	%
Coremark	Тест производительности Coremark	6 740 566	10,41
Dhrystone	Тест производительности Dhrystone	4 470 429	6,90
FFT	"Бабочка" Фурье двойной точности	5 732 961	8,85
Fibonacci	Поиск N -го числа Фибоначчи	5 607 102	8,66
gZip	Алгоритм архиватора gzip	6 408 230	9,89
Hanoi	Алгоритм перемещения "Ханойской башни"	4 998 254	7,72
Heapsort	Алгоритм пирамидальной сортировки	5 325 662	8,22
MatrixMult	Умножение вещественных матриц двойной точности различными алгоритмами	9 832 118	15,18
Queens	Решение задачи о расстановке N ферзей на шахматной доске размером $N \times N$	13 639 321	21,06
Whetstone	Тест производительности вещественной арифметики Whetstone	2 012 127	3,11
Всего		64 766 630	100,00

сказаний при выполнении кода ОС) и наиболее хорошо подходили для целей исследовательского тестирования.

Анализ полученных результатов

В первую очередь, для анализа работоспособности блока предсказания были промоделированы схемы статического и динамического предсказаний, а также (для сравнения) работа процессора без предсказаний, когда переход считается никогда не выполняющимся. Точность предсказания определялась как отношение правильно предсказанных инструкций ветвления к их общему числу. Для случая, когда предсказание было выключено, все инструкции ветвления считались "не предсказанными", и точность работы в этом случае является отношением невыполненных инструкций перехода к их общему числу. Значение IPC есть отношение числа выполненных команд (без учета отмененных из-за неправильного предсказания) к длительности всего теста в тактах.

Полученные результаты представлены на рис. 5. На основании этих данных можно говорить о том, что реализованная схема динамического предсказания работает: точность предсказания (по сравнению со статическим методом) выросла на 36 %, что дало увеличение IPC на 5 %. Небольшой (относительно увеличения точности) рост IPC объясняется, в первую очередь, наличием тестов вещественной арифметики (см. рис. 4 на третьей стороне обложки), в которых длительность выполнения веществен-

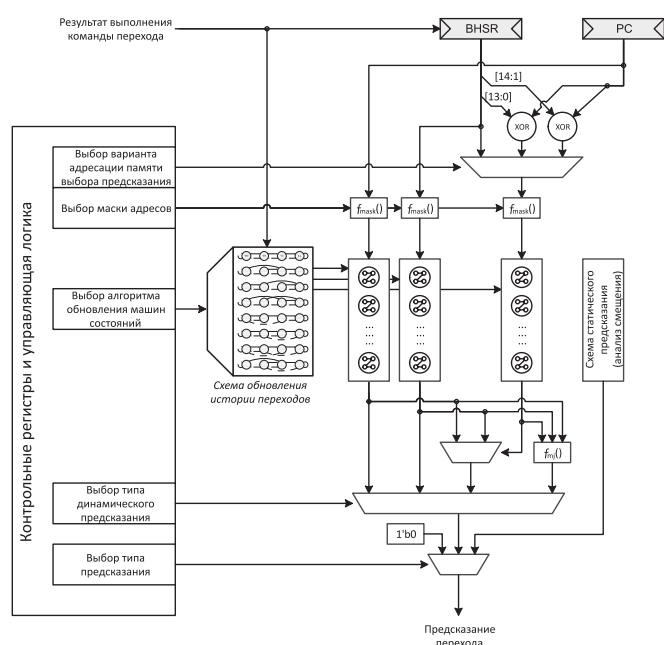


Рис. 3. Схема блока динамического предсказания с управляющей логикой, использовавшаяся при моделировании

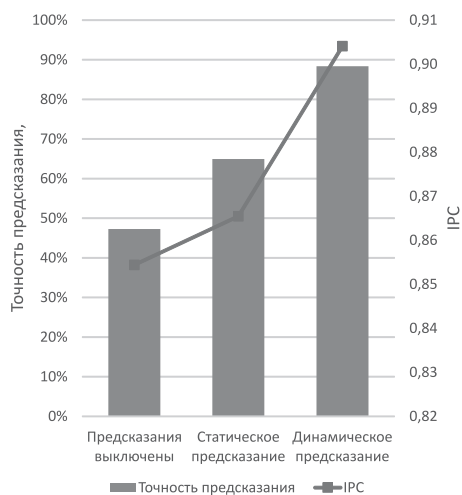


Рис. 5. Точность статического и динамического предсказаний

ных команд вносит существенный вклад в длительность всего теста и мало зависит от точности предсказаний. Вместе с тем, это показатель того, насколько сложно добиться реального увеличения производительности.

Для анализа собственно схемы динамического предсказания были промоделированы различные виды ее реализации: две схемы одноуровневого предсказания, различающиеся способом адресации памяти истории переходов (biModal) и (gShare), схема мажоритарного предсказания (majority) и исходная схема с использованием памяти выбора предсказания (choice) (см. рис. 1 и рис. 6). Полученные результаты коррелируют с известными из литературы [12, 14, 15]: одноуровневое предсказание имеет наименьшую точность, требуя при этом только одного банка памяти. Мажоритарная схема работает несколько лучше, однако требует уже три банка памяти, но в этом случае нужно реализовать схему с выбором предсказания, которая дает лучший результат при тех же аппаратных затратах.

Влияние размера памяти на точность предсказания для этих четырех видов динамического предсказания показано на рис. 7. Размер памяти с помощью маски ограничивался от 16 байт (размера строки физического блока памяти) до полного размера в 4 Кбайт. На графике видно, что точность схемы с выбором предсказания выше при любом размере памяти, и не растет при увеличении размера памяти выше 512 байт. Некоторое уменьшение точности при размерах памяти свыше 512 байт объясняется потерями на накопление истории, так как для большего объема памяти требуется больше времени на сбор истории переходов, но оно еще не компенсируется увеличением точности работы схемы выбора.

В целом, собранные данные на рис. 6 и 7 коррелируют с приводимыми в источниках [16, 17] и подтверждают правильность реализации как самих схем, так и управляющей логики, которой эти схемы выбирались.

Основываясь на этих результатах, для дальнейшего анализа была выбрана схема предсказания с выбором (choice). В исходной реализации память выбора предсказания адресовалась значением, полученным сложением по модулю 2 (XOR) программного счетчика и сдвигового регистра глобальной истории переходов (branch history shift register, BHSR). Кроме такой реализации в качестве адреса можно использовать значение самого программного счетчика и регистра BHSR или изменить функцию их комбинирования (например, для функции XOR не учитывать значение последнего перехода). Такой анализ должен показать, можно ли улучшить реализацию схемы предсказания изменением адресации ее памяти. По результатам моделирования, приведенным на рис. 8, лучший результат показала схема, в которой в качестве адреса памяти выбора предсказания используется значение программного счетчика. Поэтому для дальнейшего использования в схеме выбран такой механизм адресации.

При описаниях схем динамического предсказания в работе [18] встречается анализ изменений алгоритма обновления двухбитного конечного автомата, лежащего в основе схемы предсказания.

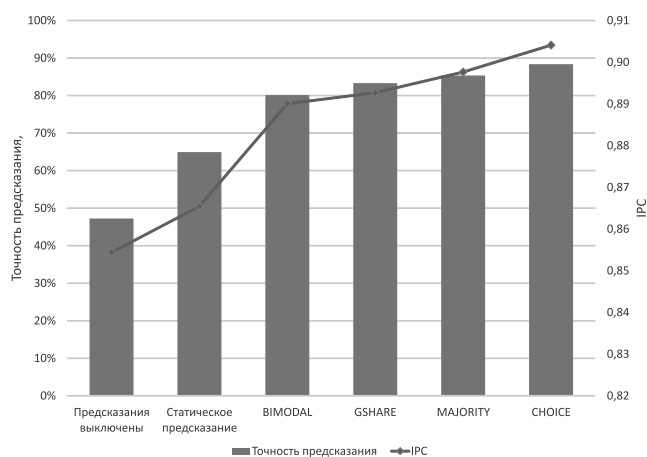


Рис. 6. Точность разных видов динамического предсказания

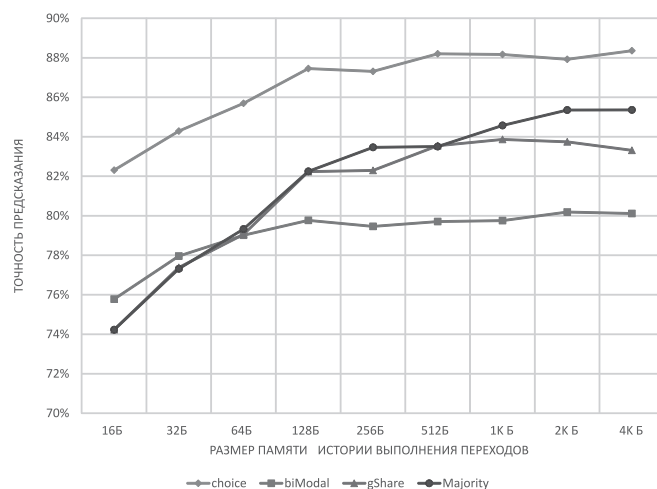


Рис. 7. Точность предсказания в зависимости от размера памяти

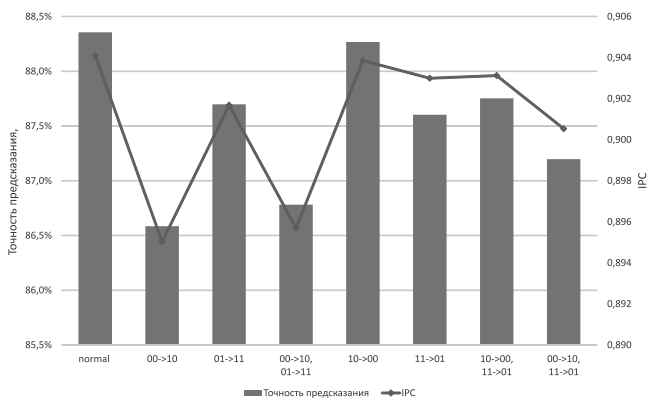


Рис. 8. Точность предсказания при различных вариантах адресации памяти выбора результата

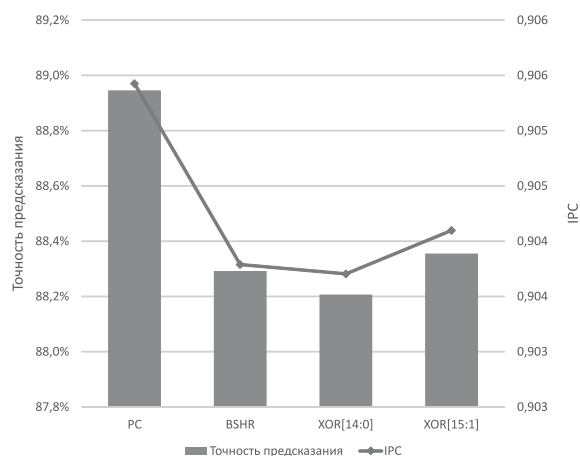


Рис. 9. Точность предсказания при различных алгоритмах обновления машин состояний

Анализ начального состояния истории не проводился, так как его невозможно контролировать в микросхеме, но статистика изменения точности предсказаний при различных алгоритмах обновления значения двухбитного конечного автомата была собрана, результаты показаны на рис. 9.

В среднем, классическая реализация двухбитного счетчика оказалась лучшей для большинства тестов, а попытка предсказывать переход как "выполняемый" после первого его реального выполнения приводила даже к худшим результатам, чем предсказание перехода "невыполняемым" после первой же невыполненной команды ветвления. Хотя были тесты, для которых изменение логики обновления истории приводило к улучшению точности (и, самое главное, к увеличению IPC), для остальных тестов эти же изменения сказывались отрицательно. Возможно, стоит проверить тесты, для которых в литературе указывается лучшая стратегия обновления истории, но это может быть результатом отдельного исследования.

Заключение

В результате проведенного исследования была обоснована корректность реализации блока динами-

ческого предсказания в микропроцессоре КОМДИВ, а также проанализировано влияние различных параметров на точность работы схемы динамического предсказания. Проведенное исследование показывает, что реализованная гибридная схема динамического предсказания является максимально производительной при данных аппаратных затратах. Единственное изменение, внесенное в схему по результатам моделирования, касалось адресации памяти выбора предсказания: оно было изменено с функции XOR на адресацию памяти счетчиком команд. Никакие изменения других параметров не приводили к увеличению точности работы схемы.

Кроме того, была собрана и проанализирована статистика по изменению производительности при применении той или иной схемы. На основании этих данных можно обоснованно выбирать для использования разные схемы динамического предсказания в случае необходимости экономии аппаратных ресурсов. Например, влияние памяти на точность предсказания ограничивается 512 байтами, т. е. в случае необходимости ее можно уменьшить без существенной потери точности работы схемы и производительности.

Также было проанализировано влияние различных факторов на работу схемы предсказания. Так, изменение вариантов адресации памяти выбора предсказания или алгоритмов обновления машин состояний истории переходов может давать эффект для определенных программ и может быть полезно при оптимизации микропроцессора для решения именно этого класса задач. Кроме этого, ожидается, что на точность будет оказывать влияние число параллельно работающих памятей и способ выбора конечного предсказания, что является темой для следующих исследований.

Список литературы

1. Huang Y., Zha Z., Chen M. and Zhang L. Moby: A Mobile Benchmark Suite for Architectural Simulators // IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). 2014. P. 45–54.
2. Intel Atom Processor 330. 2008. URL: http://ark.intel.com/products/35641/Intel-Atom-Processor-330-1M-Cache-1_60-GHz-533-MHz-FSB
3. Cortex-A Series Programmer's Guide, rev. 4.0, 2014. URL: <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.den0013d/index.html>
4. Halfhill T. R. P5600 Extends MIPS Performance. Microprocessor Report, 2013.
5. Бобков С. Г. Архитектура перспективных высокопроизводительных микропроцессоров // Программные продукты и системы. 2012. № 3. С. 63–68.
6. Бобков С. Г. Импортзамещение элементной базы вычислительных систем // Вестник Российской академии наук. 2014. Т. 84, № 11. С. 1010–1016.
7. The microarchitecture of Intel, AMD and VIA CPUs: An optimization guide for assembly programmers and compiler makers. URL: <http://www.agner.org/optimize/microarchitecture.pdf> (последнее изменение 19.02.2014).
8. Kessler R. E. The Alpha 21264 Microprocessor // IEEE Micro. 1999. Vol. 19, No. 2. P. 24–36.
9. Compaq Computer Corp., Alpha 21264 Microprocessor Hardware Reference Manual, 1999.

10. **Tendler J. M., Dodson J. S., Fields J. S., Le H., Sinharoy B.** POWER4 System Microarchitecture // *IBM Journal of Research and Development*. 2002. Vol. 46, No 1. P. 5–25.

11. **Sinharoy B., Kalla R. N., Tendler J. M., Eickemeyer R. J., and Joyner J. B.** POWER5 System Microarchitecture // *IBM Journal of Research and Development*. 2005. Vol. 49, No 4/5. P. 505–521.

12. **Michaud P., Seznec A., and Uhlig R.** Trading conflict and capacity aliasing in conditional branch predictors // *Proceedings of the 24th Annual International Symposium on Computer Architecture (ISCA-97)*. 1997. P. 292–303.

13. **Seznec A., Felix S., Krishnan V., Sazeides Y.** Design Tradeoffs for the Alpha EV8 Conditional Branch Predictor // *29th IEEE-ACM International Symposium on Computer Architecture (ISCA-02)*. 2002. P. 295–306.

14. **McFarling S.** Combining Branch Predictors // *Technical Report TN-36, Digital Western Research Laboratory*, 1993.

15. **Eden A. N. and Mudge T. N.** The YAGS Branch Prediction Scheme // *31st annual ACM/IEEE international symposium on Microarchitecture*. 1998. P. 69–77.

16. **Yeh T.-Y. and Partt Y. N.** A Comparison of Dynamic Branch Predictors that Use Two Levels of Branch History // *24th annual ACM/IEEE International Symposium on Microarchitecture*. 1993. P. 257–266.

17. **Stark J., Evers M., and Patt Y. N.** Variable Length Path Branch Prediction // *8th international conference on Architectural support for programming languages and operating systems*. 1998. P. 170–179.

18. **Shen J. P., Lipasti M. H.** *Modern Processor Design: Fundamentals of Superscalar Processors* // Waveland Press, Inc., USA, 2013.

M. E. Barskikh¹, Head of section, e-mail: barskikh@cs.niisi.ras.ru,

S. G. Bobkov^{1, 2}, Doctor of Science, Deputy Director, e-mail: bobkov@cs.niisi.ras.ru

¹ Russian Academy of Sciences, Scientific Research Institute for System Analysis

² National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)

Investigation of the Influence of Dynamic Branch Prediction to Perspective Microprocessors Performance from NIISI RAS

The effect of different dynamic branch prediction methods on the perspective microprocessors performance with RISC architecture KOMDIV is investigated. The simulation was performed on the System-on-Chip RTL-model with system controller and CPU core, which gives cycle-accuracy results. The dynamic prediction block was made configurable to be able to control the various prediction schemes. For the test cases a well-known benchmarks (Coremark, Dhrystone, etc.), and a real application programs (gZip, MatrixMult and other) have been used. The comparative studies of various effects of dynamic branch prediction schemes were conducted with the analysis of hardware consumption. Based on these data, choice of combined prediction scheme (with memory to prediction choice) for use is substantiated in the article. Besides, size and memory addressing function influence were collected and analyzed for the prediction accuracy. Based on this data modification in the memory addressing algorithm were amended: program counter has been selected directly to select memory instead of XOR function. Additional analysis of 2-bit FSM update algorithms showed the possibility of improving the accuracy and, most importantly, the performance for certain tests only, but, in general, the classic algorithm 2-bit counter is the best choice. This research confirmed the gain from dynamic branch prediction used in processors with a short pipeline and architecture KOMDIV for the feasible application of it for the serial chips.

Keywords: superscalar processor, dynamic branch prediction, simulation

References

1. **Huang Y., Zha Z., Chen M. and Zhang L.** Moby: A Mobile Benchmark Suite for Architectural Simulators. *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2014, pp. 45–54.

2. **Intel Atom Processor 330**, 2008. URL: http://ark.intel.com/products/35641/Intel-Atom-Processor-330-1M-Cache-1_60-GHz-533-MHz-FSB

3. **Cortex-A Series Programmer's Guide**, rev. 4.0. 2014. URL: <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.den0013d/index.html>

4. **Halfhill T. R.** P5600 Extends MIPS Performance. *Microprocessor Report*, 2013.

5. **Bobkov S. G.** Архитектура перспективных высокопроизводительных микропроцессоров. *Программные продукты и системы*, 2012, no. 3, pp. 63–68.

6. **Bobkov S. G.** Importozameshchenie elementnoi bazy vychislitel'nykh sistem. *Vestnik Rossiiskoi Akademii Nauk*, 2014, vol. 84, no. 11, pp. 1010–1016.

7. **The microarchitecture of Intel, AMD and VIA CPUs: An optimization guide for assembly programmers and compiler makers.** URL: <http://www.agner.org/optimize/microarchitecture.pdf> (last update 19.02.2014).

8. **Kessler R. E.** The Alpha 21264 Microprocessor. *IEEE Micro*, 1999, vol. 19, no. 2, pp. 24–36.

9. **Compaq Computer Corp.**, *Alpha 21264 Microprocessor Hardware Reference Manual*, 1999.

10. **Tendler J. M., Dodson J. S., Fields J. S., Le H., Sinharoy B.** POWER4 System Microarchitecture. *IBM Journal of Research and Development*, 2002, vol. 46, no. 1, pp. 5–25.

11. **Sinharoy B., Kalla R. N., Tendler J. M., Eickemeyer R. J., and Joyner J. B.** POWER5 System Microarchitecture. *IBM Journal of Research and Development*, 2005, vol. 49, no. 4/5, pp. 505–521.

12. **Michaud P., Seznec A., and Uhlig R.** Trading conflict and capacity aliasing in conditional branch predictors. *Proceedings of the 24th Annual International Symposium on Computer Architecture (ISCA-97)*, 1997, pp. 292–303.

13. **Seznec A., Felix S., Krishnan V., Sazeides Y.** Design Tradeoffs for the Alpha EV8 Conditional Branch Predictor. *29th IEEE-ACM International Symposium on Computer Architecture (ISCA-02)*, 2002, pp. 295–306.

14. **McFarling S.** Combining Branch Predictors. *Technical Report TN-36, Digital Western Research Laboratory*, 1993.

15. **Eden A. N. and Mudge T. N.** The YAGS Branch Prediction Scheme. *31st annual ACM/IEEE international symposium on Microarchitecture*, 1998, pp. 69–77.

16. **Yeh T.-Y. and Partt Y. N.** A Comparison of Dynamic Branch Predictors that Use Two Levels of Branch History". *24th annual ACM/IEEE International Symposium on Microarchitecture*, 1993, pp. 257–266.

17. **Stark J., Evers M., and Patt Y. N.** Variable Length Path Branch Prediction. *8th international conference on Architectural support for programming languages and operating systems*, 1998, pp. 170–179.

18. **Shen J. P., Lipasti M. H.** *Modern Processor Design: Fundamentals of Superscalar Processors*. Waveland Press, Inc., USA, 2013.

СТРУКТУРНЫЙ СИНТЕЗ И ПРИНЯТИЕ РЕШЕНИЙ STRUCTURAL SYNTHESIS AND DECISION MAKING

УДК 51-74

А. В. Вишнеков, д-р техн. наук, проф., проф. каф.,

Е. М. Иванова, доц., доц. каф., e-mail: avishnekov@hse.ru, emivanova@hse.ru

Департамент компьютерной инженерии Московского института электроники и математики
Национального исследовательского университета "Высшая школа экономики", Москва

К вопросу об автоматизации процедур определения победителей конкурсов научно-технических проектов и стартапов

Рассматриваются вопросы построения автоматизированных систем для определения победителей конкурсов научно-технических проектов на основе применения методов поддержки принятия решений. Предлагается интегрированный подход, включающий процедуры методов аналитических иерархий и групповых методов экспертной оценки многокритериальных альтернатив. Приводятся примеры расчетов, иллюстрирующие данный подход.

Ключевые слова: конкурс проектов, автоматизированные системы, поддержка принятия решений

Введение

Сегодня существует множество возможностей для научных работников и молодых специалистов по участию в различных ведомственных, отраслевых, региональных, российских и международных конкурсах исследовательских проектов. Перед организаторами подобных мероприятий стоит важная задача по отбору среди поступивших заявок наиболее перспективных, вложение в которые будут экономически оправданы и со временем окупятся.

На официальных порталах организаторов конкурсов научно-технических проектов или стартапов размещены материалы по порядку организации, проведения мероприятий и отбора конкурсантов: нормативная документация, сведения о многоэтапном порядке проведения экспертных процедур, о группах экспертов и работе экспертных советов [1–2]. Можно сделать вывод, что процедура отбора заявок достаточно сложная, многоэтапная и трудоемкая, а иногда и недостаточно прозрачна.

Авторы настоящей статьи предлагают пути автоматизации экспертных процедур выбора лучших проектов/заявок на основе использования методов поддержки принятия решений.

С точки зрения теории принятия решений задача выбора одного наиболее рационального решения из многих предложенных альтернатив сводится к анализу предметной области, выбору и оцениванию важности критериев отбора и сравнению альтернатив по выбранным критериям. Для формальной постановки задачи принятия решения по указанной проблеме важно:

- сформировать перечень альтернатив — таковыми будут поданные заявки для участия в том или ином конкурсе;

- сформировать перечень критериев отбора альтернатив, основываясь на представленной нормативной конкурсной документации;
- определить шкалы численных или лингвистических оценок по выбранным критериям;
- выбрать и обосновать метод теории принятия решения, наиболее подходящий для определения победителей конкурса проектов.

1. Постановка задачи

Перечни критериев отличаются от конкурса к конкурсу и по количеству, и по номенклатуре. Однако можно отметить, что большинство критериев отбора во всех случаях носят лингвистический (описательный) характер, а не численный. В связи с этим, проанализировав методы поддержки принятия решений, можно выделить как приемлемые: метод ЗАПРОС [3], метод аналитических иерархий [3], метод идеальной точки [4], метод кластеризации экспертных оценок альтернатив [5].

В методе ЗАПРОС построение правила упорядочения многокритериальных альтернатив связано с применением достаточно трудоемких интерактивных обучающих процедур. Данный метод позволяет наиболее детально учесть взаимосвязь отдельных критериев. При этом требуется сложная процедура решения вопроса проверки информации на непротиворечивость.

Метод аналитических иерархий применим при небольшом числе альтернатив, так как требует заполнение экспертами или лицом, принимающим решение (ЛПР), матриц попарного сравнения всех альтернатив по всем критериям, что при большом

числе альтернативных проектов вызывает значительные затруднения или становится невозможным.

Метод идеальной точки чаще всего применяется, когда все критерии отбора имеют одинаковую важность. Если при отборе заявок это условие выполнено, то применение данного метода оправдано, но в общем случае критерии могут иметь различную важность. Метод работает при условии, что все оценки носят численный, но не лингвистический характер.

Метод кластеризации экспертных оценок альтернатив требует трудоемкой процедуры многокритериальной экспертной оценки каждой альтернативы, что ограничивает возможность применения метода при большом числе альтернатив.

Так как каждый из рассмотренных методов имеет ограничения или сложности в применении, особенно при наличии лингвистических оценок на шкалах критериев, то предлагается интегрированный подход, построенный на основе применения процедур принятия решений, используемых в методах аналитических иерархий, групповых методов экспертной оценки многокритериальных альтернатив [6] — метод предпочтений, метод ранга, методы согласования групповых решений с использованием функций или отношений предпочтения ЛПР [4].

Рассмотрим в качестве небольшого примера процедуру отбора заявок для конкурса научных проектов, выполняемых молодыми учеными [1]. Приведем используемый перечень критериев для конкурсного отбора:

- научная значимость решаемых в проекте задач;
- актуальность заявленной темы исследования;
- степень новизны методов и подходов;
- четкость изложения цели и задач;
- наличие у руководителя проекта статей в рецензируемых научных журналах;
- наличие у руководителя проекта научных наград и премий.

При использовании этих критериев жюри оценивает все поступившие на конкурс заявки.

Таким образом, для формализации постановки задачи мы имеем три пункта из названных четырех: сформирован перечень альтернатив/заявок, сформирован перечень критериев, выбран метод решения. Остается определить порядковые шкалы численных или лингвистических оценок по выбранным критериям.

Предположим, что каждый проект получит по каждому критерию некоторую лингвистическую оценку, определяемую группой экспертов. Поскольку таких оценок может быть множество, то для формализации процедуры оценивания воспользуемся алгоритмом метода ЗАПРОС, где предлагается свести все оценки к нескольким возможным вариантам — градациям на оценочных шкалах. Допустим, экспертами предложены следующие градации на шкалах оценок альтернатив по каждому критерию.

Критерий К1. Научная значимость решаемых в проекте задач.

Градация шкалы:

- К1.1. Уникальные исследования мирового или государственного значения.
- К1.2. Высокое значение для конкретной отрасли Российской Федерации.
- К1.3. Имеет ведомственное значение.
- К1.4. Трудно оценить значимость проекта.

Критерий К2. Актуальность заявленной темы исследования.

Градация шкалы:

- К2.1. Исследования актуальны на сегодняшний день.
- К2.2. Исследования актуальны в перспективе.
- К2.3. Обоснование актуальности отсутствует.
- К2.4. Актуальность низкая.

Критерий К3. Степень новизны методов и подходов.

Градация шкалы:

- К3.1. Изложенные методы и подходы полностью новые, что подтверждено результатами патентных исследований, публикаций, литературных источников.
- К3.2. Изложенные методы и подходы новые, но подтверждение тому отсутствует.
- К3.3. Часть из предложенных методов и подходов имеют научную новизну.
- К3.4. Используются хорошо известные методы и подходы.

Критерий К4. Четкость изложения цели и задач.

Градация шкалы:

- К4.1. Цели и задачи сформулированы четко и понятно.
- К4.2. Отдельные задачи сформулированы нечетко.
- К4.3. Нечеткость или размытость формулировок при постановке задач.
- К4.4. Описание цели или задач отсутствует.

Критерий К5. Наличие у руководителя проекта статей в рецензируемых научных журналах.

Градация шкалы:

- К5.1. Наличие ряда статей в отечественных и зарубежных рецензируемых научных журналах.
- К5.2. Наличие 1—2 статей в зарубежных и ряда статей в отечественных рецензируемых научных журналах.
- К5.3. Наличие ряда статей в отечественных рецензируемых научных журналах.
- К5.4. Отсутствие публикаций в рецензируемых научных журналах.

Критерий К6. Наличие у руководителя проекта научных наград и премий.

Градация шкалы:

- К6.1. Имеются государственные награды или премии.
- К6.2. Имеются отраслевые награды или премии.
- К6.3. Имеются ведомственные награды или премии.
- К6.4. Награды или премии отсутствуют.

2. Применение методов поддержки принятия решений для конкурсного отбора

Группа экспертов принимает коллективное решение о номенклатуре и градациях шкал лингвистических оценок по каждому критерию и о степени значимости каждой из лингвистических оценок.

При первичном отборе заявок (альтернатив) при их большом числе можно исключить из рассмотрения альтернативы с наихудшей оценкой (в рассмотренном выше примере — с оценкой K_{l4} , $l = 1, \dots, 6$) по одному или нескольким критериям. Это позволит значительно сократить перечень альтернатив. В нашем случае исключению могут подлежать проекты со следующими оценками:

K1.4. Трудно оценить научную значимость проекта.

K2.4. Низкая актуальность заявленной темы исследования.

K3.4. Используются хорошо известные методы и подходы.

K4.4. Нечеткость или размытость формулировок при постановке задач.

K5.4. Отсутствие у руководителя проекта публикаций в рецензируемых научных журналах.

K6.4. Отсутствие у руководителя проекта научных наград и премий.

Далее для определения численных оценок градаций на шкалах критериев предлагается применить процедуру, аналогичную назначению весов критериев в методе аналитических иерархий. Эксперты или ЛПР задают относительную важность градаций (лингвистических оценок) на порядковых шкалах критериев. Для этого ЛПР следует указать, в какой степени одна из градаций/оценок предпочтительней другой. Степень предпочтительности может выражаться числами: равная значимость — 1, умеренно превосходит — 3, умеренно уступает — $1/3$, существенно превосходит — 5, существенно уступает — $1/5$, значительно превосходит — 7, значительно уступает — $1/7$, очень сильно превосходит — 9, очень сильно уступает — $1/9$ [3].

Пусть, например, выбраны следующие оценки степени превосходства градаций/лингвистических оценок, отраженные в m_l -мерных квадратных матрицах (рис. 1) Z^{Kl} попарного сравнения оценок с номерами i и j по каждому критерию Kl . Здесь m_l — число численных оценок по каждому критерию Kl после исключения наихудших оценок по

$$Z^1 = \begin{pmatrix} 1 & 3 & 9 \\ 1/3 & 1 & 7 \\ 1/9 & 1/7 & 1 \end{pmatrix}; \quad Z^2 = \begin{pmatrix} 1 & 3 & 5 \\ 1/3 & 1 & 3 \\ 1/5 & 1/3 & 1 \end{pmatrix}; \quad Z^3 = \begin{pmatrix} 1 & 3 & 7 \\ 1/3 & 1 & 3 \\ 1/7 & 1/3 & 1 \end{pmatrix};$$

$$Z^4 = \begin{pmatrix} 1 & 3 & 7 \\ 1/3 & 1 & 3 \\ 1/7 & 1/3 & 1 \end{pmatrix}; \quad Z^5 = \begin{pmatrix} 1 & 3 & 9 \\ 1/3 & 1 & 5 \\ 1/9 & 1/5 & 1 \end{pmatrix}; \quad Z^6 = \begin{pmatrix} 1 & 3 & 9 \\ 1/3 & 1 & 7 \\ 1/9 & 1/7 & 1 \end{pmatrix}.$$

Рис. 1. Матрицы попарного сравнения лингвистических оценок альтернатив на порядковых шкалах критериев

каждому критерию. Поскольку число таких оценок по каждому критерию может быть различным, то и параметры m_1, m_2, \dots могут различаться. В нашем примере $m_1 = m_2 = \dots = m_6 = 3$.

В дальнейшем эти матрицы можно будет использовать для сравнительной оценки альтернативных заявок по всем рассмотренным критериям.

Если ЛПР затрудняется в определении подобных показателей, то следует пригласить группу экспертов (возможно жюри конкурса) для организации экспертной процедуры оценивания методом ранга [6] или методом предпочтений [6] и последующего согласования групповых решений с использованием функций или отношений предпочтения ЛПР [4].

Процедура применения метода ранга и согласования групповых решений будет рассмотрена ниже (см. раздел 3).

Рассмотрим пример конкурсного отбора заявок, который проводит ЛПР.

Пусть поступило 10 заявок (альтернативы A_1, \dots, A_{10}). После работы экспертов были сформированы лингвистические оценки каждой из альтернатив по каждому критерию (табл. 1).

При первичной оценке заявок из табл. 1 видно, что имеется 7 заявок, получивших наихудшую оценку по одному или нескольким критериям (закрашенные клетки таблицы). Тогда согласно предложенной методике эти заявки считаются не прошедшими конкурсный отбор и выбывают из дальнейшего рассмотрения, а новая сравнительная таблица значительно сократится (табл. 2).

Далее составим матрицы взаимного предпочтения альтернатив (рис. 2) с использованием значений относительной важности градаций на шкалах кри-

Таблица 1

Первичные лингвистические оценки альтернатив

Альтернатива	Критерий Kl					
	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$
A1	K1.1	K2.1	K3.3	K4.3	K5.2	K6.4
A2	K1.2	K2.2	K3.1	K4.2	K5.1	K6.2
A3	K1.3	K2.2	K3.4	K4.4	K5.2	K6.2
A4	K1.1	K2.1	K3.2	K4.2	K5.1	K6.1
A5	K1.3	K2.3	K3.3	K4.3	K5.4	K6.3
A6	K1.4	K2.1	K3.2	K4.2	K5.1	K6.2
A7	K1.3	K2.3	K3.1	K4.4	K5.2	K6.2
A8	K1.4	K2.4	K3.3	K4.1	K5.2	K6.1
A9	K1.2	K2.2	K3.1	K4.1	K5.1	K6.2
A10	K1.2	K2.1	K3.4	K4.3	K5.2	K6.2

Таблица 2

Сокращенная таблица лингвистических оценок альтернатив

Альтернатива	Оценки альтернатив по критерию Kl					
	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$
A2	K1.2	K2.2	K3.1	K4.2	K5.1	K6.2
A4	K1.1	K2.1	K3.2	K4.2	K5.1	K6.1
A9	K1.2	K2.2	K3.1	K4.1	K5.1	K6.2

$$\begin{aligned}
 x^1 &= \begin{pmatrix} 1 & 1/3 & 1 \\ 3 & 1 & 3 \\ 1 & 1/3 & 1 \end{pmatrix}; & x^2 &= \begin{pmatrix} 1 & 1/3 & 1 \\ 3 & 1 & 3 \\ 1 & 1/3 & 1 \end{pmatrix}; & x^3 &= \begin{pmatrix} 1 & 3 & 1 \\ 1/3 & 1 & 1/3 \\ 1 & 3 & 1 \end{pmatrix}; \\
 x^4 &= \begin{pmatrix} 1 & 1 & 1/3 \\ 1 & 1 & 1/3 \\ 3 & 3 & 1 \end{pmatrix}; & x^5 &= \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}; & x^6 &= \begin{pmatrix} 1 & 1/3 & 1 \\ 3 & 1 & 3 \\ 1 & 1/3 & 1 \end{pmatrix}.
 \end{aligned}$$

Рис. 2. Матрицы попарного сравнения альтернатив

териев, представленных матрицами на рис. 1. Зная степень превосходства одной градации над другой на шкалах критериев, нетрудно задать степень превосходства одной альтернативы над другой, имеющих в качестве оценки эти градации. Степень превосходства одной альтернативы над другой по рассматриваемому критерию численно равна степени превосходства градации, соответствующей оценке первой альтернативы, над второй.

Для пояснения рассмотрим пример формирования значения в строке $i = 1$ и столбце $j = 2$ матрицы $X^{kl} = 1$ (рис. 2). Число строк и столбцов в матрицах взаимного предпочтения альтернатив равно числу сравниваемых альтернатив из табл. 2 (в нашем случае — 3). Данная матрица показывает сравнение трех выбранных проектов (A2, A4, A9) по критерию K1. Элементом x_{12} является степень превосходства первой альтернативы (A2) над второй альтернативой (A4). Согласно табл. 2 по критерию K1 они имеют оценки K1.2 (A2) и K1.1 (A4). Степень превосходства градации K1.2 над K1.1 (см. матрицу Z^1 на рис. 1) равна $x_{12} = z_{21} = 1/3$ — умеренно уступает.

Числовые веса V_i^{kl} указанных лингвистических оценок можно вычислить по приведенным ниже формулам, аналогично процедурам, применяемым в методе аналитических иерархий. Здесь C_i^{kl} — цена лингвистической оценки i ($i = 1, \dots, 3$) по каждому критерию kl , S_{kl} — сумма цен лингвистических оценок.

$$C_i^{kl} = k \sqrt[k]{\prod_{j=1, \dots, 3} x_{ij}}; S_{kl} = \sum_{i=1}^3 C_i^{kl}; V_i^{kl} = C_i^{kl} / S_{kl}$$

В результате расчетов для нашего примера получим следующие числовые значения:

$$V_1^{K1} = 0,20; V_2^{K1} = 0,60; V_3^{K1} = 0,20;$$

$$V_1^{K2} = 0,20; V_2^{K2} = 0,60; V_3^{K2} = 0,20;$$

$$V_1^{K3} = 0,43; V_2^{K3} = 0,14; V_3^{K3} = 0,43;$$

$$V_1^{K4} = 0,25; V_2^{K4} = 0,25; V_3^{K4} = 0,50;$$

$$V_1^{K5} = 0,33; V_2^{K5} = 0,33; V_3^{K5} = 0,33;$$

$$V_1^{K6} = 0,20; V_2^{K6} = 0,60; V_3^{K6} = 0,20.$$

Та же экспертная группа или ЛППР может задать значимость (вес) каждого критерия для оценки

проекта в целом. Этот вес может также быть рассчитан аналогично процедурам, применяемым в методе аналитических иерархий для взаимного попарного сравнения критериев, либо путем экспертной оценки на основе метода предпочтений или методом ранга. Подобная методика была рассмотрена авторами в работе [7]. Допустим, получены следующие веса критериев: $W_{K1} = 0,26$; $W_{K2} = 0,20$; $W_{K3} = 0,10$; $W_{K4} = 0,17$; $W_{K5} = 0,12$; $W_{K6} = 0,15$.

Для всех поступивших заявок (альтернатив A_r , для $r = 1, \dots, kr$) экспертная комиссия (жюри) выбирает одну из лингвистических оценок по каждому критерию, которая однозначно соответствует числовой оценке V_i^{kl} . Обозначим O_{Ar}^{kl} оценку альтернативы A_r по критерию kl . Зная вес каждого критерия W_{kl} и оценку O_{Ar}^{kl} альтернативы, по этому критерию можно вычислить ценность каждой альтернативы для реализации проекта по формуле

$$U_{Ar} = \sum_{l=1}^6 O_{Ar}^{kl} W_{kl} \quad (1)$$

Значит, можно проранжировать (расставить по важности) все поступившие заявки/альтернативные проекты.

По формуле (1) рассчитаем ценность каждой альтернативы: $U_{A2} = 0,25$; $U_{A4} = 0,46$; $U_{A9} = 0,29$. Отсюда можно сделать вывод, что наилучшей и достойной поддержки является заявка A4 (первое место), а далее по мере убывания важности заявки A9 (второе место) и A2 (третье место).

Все рассмотренные в данном разделе алгоритмы и вычисления легко могут быть запрограммированы, и вся процедура конкурсного отбора автоматизирована.

3. Процедура экспертного оценивания и согласования групповых решений

Процедуру согласования экспертных оценок проводит ЛППР, и эта процедура также может быть автоматизирована. Рассмотрим те же данные, что и в предыдущем примере (см. табл. 1, 2). Число сравниваемых по каждому l -му критерию лингвистических оценок $j = m_l$ (в нашем примере $m_1 = m_2 = \dots = m_6 = 3$), т. е. $j = 1, \dots, 3$. Число критериев равно 6, т. е. $l = 1, \dots, 6$. Число экспертов равно 4, т. е. $i = 1, \dots, 4$.

Рассмотрим небольшой численный пример применения экспертной процедуры согласования групповых решений на основе метода ранга для определения числовых оценок градаций на шкалах критериев. В этом случае каждый из экспертов предлагает свои собственные числовые оценки (от 1 до 10) по всем критериям, соответствующие, по его мнению, каждой из лингвистических оценок — градаций (табл. 3).

Таблица 3

Экспертные числовые оценки градаций на шкалах критериев

Лингвистические оценки	Числовые экспертные оценки экспертов Э1-Э4			
	Э1	Э2	Э3	Э4
K1.1	10	9	10	8
K1.2	8	7	7	6
K1.3	3	6	2	3
K2.1	8	10	10	9
K2.2	4	6	9	8
K2.3	1	1	2	7
K3.1	10	8	9	9
K3.2	7	8	6	6
K3.3	6	8	2	2
K4.1	8	7	9	7
K4.2	7	6	8	6
K4.3	5	3	4	2
K5.1	10	9	9	8
K5.2	9	6	7	5
K5.3	7	5	3	3
K6.1	4	7	10	8
K6.2	3	6	7	6
K6.3	2	1	2	4

$$Y^1 = \begin{pmatrix} 10 & 8 & 3 \\ 9 & 7 & 6 \\ 8 & 6 & 3 \end{pmatrix}; \quad Y^2 = \begin{pmatrix} 8 & 4 & 1 \\ 10 & 6 & 1 \\ 9 & 8 & 7 \end{pmatrix}; \quad Y^3 = \begin{pmatrix} 10 & 7 & 6 \\ 8 & 8 & 8 \\ 9 & 6 & 2 \\ 8 & 6 & 4 \end{pmatrix};$$

$$Y^4 = \begin{pmatrix} 8 & 7 & 5 \\ 7 & 6 & 3 \\ 9 & 8 & 4 \\ 7 & 6 & 2 \end{pmatrix}; \quad Y^5 = \begin{pmatrix} 10 & 9 & 7 \\ 9 & 6 & 5 \\ 9 & 7 & 3 \\ 8 & 5 & 3 \end{pmatrix}; \quad Y^6 = \begin{pmatrix} 4 & 3 & 2 \\ 7 & 6 & 1 \\ 10 & 7 & 2 \\ 8 & 5 & 2 \end{pmatrix}.$$

Рис. 3. Матрицы числовых экспертных оценок для каждой из лингвистических оценок альтернатив по каждому критерию K_l

Далее все выставленные экспертами числовые оценки сводятся в оценочные матрицы Y^{Kl} по каждому из критериев K_l (на рис. 3 общее число матриц совпадает с числом критериев и в нашем случае равно 6). В каждой из матриц число строк совпадает с числом экспертов (в нашем случае — 4), а число столбцов — с числом m_l лингвистических оценок по каждому критерию K_l (в нашем случае — 3).

Элементом матрицы является величина y_{ij} , где индекс i соответствует номеру эксперта, а индекс j — номеру лингвистической оценки K_{lj} , рассчитаем веса каждой из лингвистических оценок V_i^{Kl} по каждому критерию.

Метод ранга предполагает, что веса указанных лингвистических оценок вычисляются по формулам [6]

$$S_j^l = \sum_{i=1}^{m_l} y_{ij}^l, \quad V_i^{Kl} = \frac{S_j^l}{\sum_{l=1}^6 S_j^l}.$$

Здесь S_j^l — суммарная экспертная оценка каждой лингвистической оценки K_{lj} по каждому l -му критерию для $l = 1, \dots, 6$.

Тогда числовые веса V_i^{Kl} указанных градаций/лингвистических оценок примут следующие значения:

$$V_1^{K1} = 0,47; \quad V_2^{K1} = 0,35; \quad V_3^{K1} = 0,18;$$

$$V_1^{K2} = 0,49; \quad V_2^{K2} = 0,36; \quad V_3^{K2} = 0,15;$$

$$V_1^{K3} = 0,43; \quad V_2^{K3} = 0,33; \quad V_3^{K3} = 0,24;$$

$$V_1^{K4} = 0,43; \quad V_2^{K4} = 0,38; \quad V_3^{K4} = 0,19.$$

$$V_1^{K5} = 0,44; \quad V_2^{K5} = 0,33; \quad V_3^{K5} = 0,22.$$

$$V_1^{K6} = 0,51; \quad V_2^{K6} = 0,37; \quad V_3^{K6} = 0,12.$$

Для согласования групповых решений и определения взаимной важности градаций/лингвистических оценок эти веса по каждому критерию отдельно наносят на числовую ось с пятью выделенными интервалами. Например, по критерию K_1 получим картину, приведенную на рис. 4.

Для получения взаимной числовой оценки двух альтернатив сопоставляют веса их лингвистических оценок, нанесенные на числовую ось. Если веса лингвистических оценок попадают в один интервал, то взаимная важность альтернатив по данному критерию оценивается как "1" — равная значимость. Если веса лингвистических оценок попадают в соседние интервалы, то их взаимная важность оценивается как "3 и 1/3" — незначительное превосходство и незначительная уступка и т. д.

Например, степень важности двух альтернативных проектов A_1 и A_2 с лингвистическими оценками $K_{1.1}$ и $K_{1.3}$ соответственно будет оценена следующим образом. Проект A_1 с лингвистической оценкой $K_{1.1}$ будет иметь более высокую относительную важность по сравнению с проектом A_2 с лингвистической оценкой $K_{1.3}$ (степень важности равна 5). Следовательно, важность проекта A_2 относительно проекта A_1 составит 1/5.

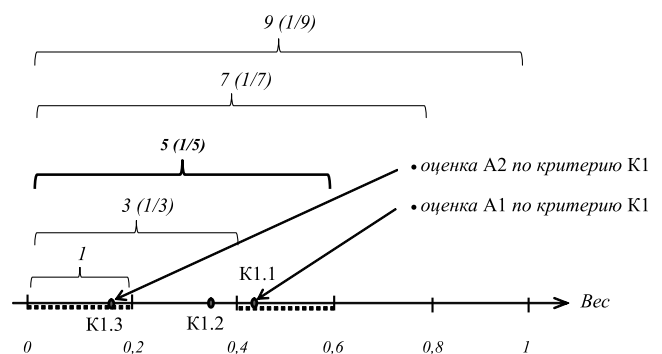


Рис. 4. Способ определения взаимной важности лингвистических оценок при согласовании групповых решений

Это позволит, как и в предыдущем примере автоматически заполнить все матрицы попарного сравнения альтернативных проектов и стартапов (см. рис. 2).

Все последующие вычисления будут аналогичны предыдущему примеру.

Заключение

Использование предложенного подхода позволяет повысить объективность принимаемых решений, связанных с определением победителей конкурсов научно-технических проектов и стартапов, сократить время и трудоемкость процедур принятия решений. Предлагаемый подход позволяет также решить вопрос проверки информации на непротиворечивость при построении решающих правил, так как использует стандартные процедуры известных методов. Для проверки непротиворечивости экспертных суждений в методе аналитических иерархий рассчитывается отношение согласованности [3]. Для проверки экспертных оценок на согласованность в методах экспертной оценки многокритериальных альтернатив, в частности в методе предпочте-

ний, рассчитывается коэффициент конкордации [6]. Для проверки экспертных оценок на согласованность в методе ранга рассчитывается дисперсия оценок экспертов [6]. Это стандартные алгоритмы, которые также могут быть легко запрограммированы.

Таким образом, предлагаемый подход позволяет автоматизировать всю процедуру принятия решений экспертами и решать задачи конкурсного отбора заявок при большом числе градаций на порядковых шкалах критериев оценки проектов.

Список литературы

1. Портал РФФИ (<http://www.rfbr.ru/rffi/ru>)
2. Интернет-портал "Наука и Инновации" (<http://www.rsci.ru>).
3. Ларичев О. И. Теория и методы принятия решений. 2-е изд., перераб. и доп. М.: Логос, 2002. 392 с.
4. Трахтенгерц Э. А. Компьютерная поддержка принятия решений. Научно-практическое издание. М.: СИНТЕГ, 1998. 376 с.
5. Вишнеков А. В., Ерохин В. В., Иванова Е. М. Верификация СНК: выбор стратегии // Нано- и микросистемная техника. 2014. № 12. С. 30–36.
6. Емельянов С. В., Ларичев О. И. Многокритериальные методы принятия решений. М.: Знание, 1985.
7. Вишнеков А. В., Ерохин В. В., Иванова Е. М. Автоматизация процедуры выбора микроконтроллера // Нано- и микросистемная техника. 2014. № 7. С. 14–21.

A. V. Vishnekov, Professor, e-mail: avishnekov@hse.ru,

E. M. Ivanova, Ass. Professor, e-mail: emivanova@hse.ru

National Research University Higher School of Economics,

Moscow Institute of Electronics and Mathematics, "Computer Engineering" department, Moscow

To the Question of Winners Definition Procedures Automation for the Scientific and Technical Projects and Startups Competition

The process of selecting a contest winner is presented in the form of a multistage decision-making procedure. At the first stage, it is necessary to generate a list of alternative projects (applications for participation in a particular contest). At the second stage a list of criteria to select the alternatives is formed. Next, it is required to determine numerical or linguistic estimation scales based on the selected criteria. It is assumed that before the contest is held several ordered gradations are determined for each criterion and estimation of each of the proposed alternative projects can be attributed to one of them. At the next stage, the problem of determining the relative importance of criteria and corresponding gradations for these criteria is solved. Then comes the procedure itself, which compares the received applications using the selected criteria and estimation scales. The last task is very labour intensive and its automation will allow to simplify greatly the work of the juries.

This paper examines the questions related to building automated systems aimed to determine the winners of scientific and technical project competitions based on the decision-making support methods. It offers an integrated approach including technologies and procedures of analytical hierarchies methods (can be used for the purpose of gradation weight on the criteria scales), group methods of expert estimation of multicriteria alternatives — the method of preferences or rank method (can be used for purposing of scales gradation on scales of criteria and for the purpose of criteria weight), methods of group decisions coordination using the preference functions or decision-maker relations (it is used for coordination of expert numerical estimates of gradation on scales of criteria). The examples of calculations illustrating this approach are given.

Keywords: projects competition, automated systems, decision-making support

References

1. Portal RFFI (<http://www.rfbr.ru/rffi/ru>)
2. Internet-portal "Наука и Инновации" (<http://www.rsci.ru>).
3. Larichev O. I. *Teoriya i metody prinjatija reshenij*. Moscow, Logos, 2002, 392 p.
4. Trahtengerc Je. A. *Komp'yuternajapodderzhka prinjatija reshenij*. Nauchno-prakticheskoe izdanie, Moscow, SINTEG, 1998, 376 p.
5. Vishnekov A. V., Erohin V. V., Ivanova E. M. Verifikacija SNK: vybor strategii, *Nano- i mikrosistemnaja tehnika*, 2014, no. 12, pp. 30–36.
6. Emel'janov S. V., Larichev O. I. *Mnogokriteriarnye metody prinjatija reshenij*, Moscow, Znanie, 1985.
7. Vishnekov A. V., Erohin V. V., Ivanova E. M. Avtomatizacija procedury vybora mikrokontrollera, *Nano- i mikrosistemnaja tehnika*, 2014, no. 7, pp. 14–21.

УДК 004.272.2

А. Л. Стемповский, д-р техн. наук, академик РАН, директор,
Д. В. Тельпухов, канд. техн. наук, зав. отделом, e-mail: nofrost@inbox.ru,
Р. А. Соловьев, канд. техн. наук, вед. науч. сотр., **М. В. Мячиков**, студент, инженер-исследователь,
Институт проблем проектирования в микроэлектронике РАН

Повышение отказоустойчивости логических схем с использованием нестандартных мажоритарных элементов

Рассмотрена актуальная задача повышения надежности комбинационных микроэлектронных схем. Предложен новый способ повышения отказоустойчивости с использованием нестандартных мажоритарных элементов (воутеров), который опирается на использование неравномерности сигнала на выходе схемы, благодаря чему появляется возможность построения более эффективных воутеров. Рассмотрен обобщенный подход внесения избыточности в схему в целях повышения надежности, а также разработана методика построения таблицы истинности оптимального мажоритарного элемента. В экспериментальной части было проведено сравнение метода с классическим трехкратным мажорированием. На частных примерах показана высокая эффективность метода в сравнении с традиционными подходами.

Ключевые слова: метод повышения надежности, отказоустойчивость, комбинационная схема, мажорирование

Введение

Интегральные схемы, используемые в вычислительных и других электронных системах, работающие под воздействием радиации и других дестабилизирующих факторов, подвержены возникновению одиночных сбоев (single event upset, SEU). Такие ошибки носят кратковременный характер и не имеют долгосрочных последствий для корректной работы интегральной схемы [1].

Исследования двадцатилетней давности констатировали необходимость защиты только регистров и модулей памяти из-за малой вероятности захвата неправильного значения сигнала регистром, а также из-за того, что комбинационная логика гораздо менее восприимчива к влиянию одиночных сбоев [2, 3]. Естественным результатом планомерной работы с тех пор явились многочисленные программные и аппаратные методы защиты последовательных элементов и модулей памяти, которыми оснащены большинство современных микропроцессоров и устройств. Однако тренд этих лет, заключающийся в росте степени интеграции, уменьшении порогового напряжения и напряжения питания, в значительной мере менял соотношение в сторону необходимости обеспечения отказоустойчивости комбинационных схем. Кроме обозначенных причин важно отметить фактор увеличения тактовых частот, увеличивающий шансы захвата ошибочных результатов, выдаваемых логическими схемами. Для

процессоров и иных микроэлектронных устройств, где последовательные элементы уже имеют встроенные механизмы защиты, комбинационные схемы быстро стали основным источником одиночных сбоев [4].

Несмотря на достигнутые успехи в разработке кодовых методов защиты, используемых при хранении, передаче, а также арифметической обработке данных [5, 6], для обеспечения требуемого уровня отказоустойчивости арифметических и логических схем до сих пор используют архаичные методы кратного резервирования [7].

Достоинством, и в то же время недостатком, традиционных мажоритарных подходов является тот факт, что они не учитывают внутренней структуры и особенностей функционирования защищаемых устройств. С одной стороны, это обеспечивает универсальность метода, позволяя применять его для любых устройств практически на всех уровнях описания схем, начиная с транзисторного представления и заканчивая логическим и микросистемным уровнем. С другой стороны, отсутствие учета различных характеристик внутреннего строения, а также вероятностного распределения сигналов на входах и выходах схемы оставляет значительное пространство для возможных усовершенствований, одному из которых посвящена настоящая работа.

В данной статье рассматривается метод мажорирования на уровне логических комбинационных

схем. Усовершенствование этого метода заключается в построении воуэтеров с более эффективной структурой. Эту структуру можно получить, проанализировав работу схемы и получив вероятностное распределение сигналов на ее выходах.

Метод оценки отказоустойчивости комбинационных схем

Для того чтобы оценивать эффективность предлагаемых подходов и иметь возможность сравнения с традиционным мажорированием, необходимо разработать адекватный критерий для оценки архитектурной устойчивости к сбоям. Рассмотрим вначале некоторые традиционные характеристики, предлагаемые в литературе. Чаще всего используются два основных параметра, вычисляемые в процессе моделирования [8].

Первый параметр характеризует архитектурную устойчивость к ошибкам и определяется как отношение числа некорректных результатов работы схемы к общему числу инжектированных ошибок:

$$T_o = \frac{I_{cr}}{T_{er}}, \quad (1)$$

где I_{cr} — число экспериментов с несовпадением результата с эталонным; T_{er} — общее число внедренных ошибок. По сути, этот параметр определяет вероятность того, что внедренная ошибка в схему приведет к искажению работы всей схемы.

Второй параметр определяет общую устойчивость схемы к ошибкам и вычисляется как отношение числа C_r корректных результатов работы схемы к общему числу T_r тестов:

$$A = \frac{C_r}{T_r}. \quad (2)$$

Эти параметры, по сути, являются комплементарными. В условиях "шума окружающей среды" параметр общей устойчивости (2) оценивает вероятность сбоя в конкретных условиях, в то время как параметр архитектурной устойчивости оценивает вероятность сбоя в пересчете на каждую внедренную ошибку, что позволяет более объективно сравнивать схемы с большими различиями в количестве вентилях в условиях высокой вероятности ошибок.

Обозначенные характеристики отказоустойчивости определены лишь в конкретных условиях для заданной модели возникновения ошибок при конкретных параметрах. Использование этих характеристик для сравнения отказоустойчивости схем затруднено из-за необходимости оценки для разных моделей ошибок на большом диапазоне изменения параметров, при которых результаты сравнения могут существенно различаться.

Оба рассматриваемых критерия отказоустойчивости являются частными случаями некоторого

общего критерия, который мы будем называть *полиномом ошибки*. В общем смысле это аналитическое выражение, характеризующее отказоустойчивость схемы в условиях "шума окружающей среды". Обозначим вероятность ошибки в одном элементе схемы p и введем характеристическую функцию множества пар векторов входного сигнала X и ошибки e :

$$E(X, e) = \begin{cases} 1, & \text{если набор } X, e \text{ приводит} \\ & \text{к ошибке на выходе;} \\ 0, & \text{иначе.} \end{cases}$$

Учитывая, что вероятность появления на входе конкретного вектора входных сигналов X длины N (в предположении равномерности всех таких наборов) равна $\frac{1}{2^N}$, а вероятность возникновения вектора

ошибки e длины M и веса $|e|$ равна $p^{|e|} (1-p)^{M-|e|}$, получаем вероятность ошибки на выходе схемы:

$$EOF(p) = \frac{1}{2^N} \sum_{X, e} E(X, e) p^{|e|} (1-p)^{M-|e|}, \quad (3)$$

где суммирование ведется по всем возможным X, e .

Полином ошибки $EOF(p)$ обобщает параметр (2) для любой вероятности ошибки на вентиле, однако нахождение его требует больших вычислительных затрат. Для реальных применений трудно себе представить ситуацию, когда необходимо обеспечивать надежность устройства при $p \rightarrow 1$. Чаще всего математическое ожидание числа ошибок в схеме близко к единице, а вероятность ошибки на вентиле стремится к нулю, а точнее $p \sim \frac{1}{M}$, где M — число элементов в схеме.

Учитывая этот факт, можно сделать вывод о том, что для корректной оценки отказоустойчивости не обязательно вычислять весь полином ошибки. Достаточно вычислить полином до определенной степени, обеспечивая допустимую погрешность на значениях p от нуля до нескольких процентов.

Для этого в выражении (3) $E(X, e)$ вычисляется только для некоторого подмножества входных X и ошибочных e наборов, а для оставшихся X, e принимается равным 0(1) для нижней (верхней) оценки полинома ошибки соответственно. Увеличивая число наборов, для которых вычисляется истинное значение $E(X, e)$, можно получать все более точные оценки и уменьшать погрешность измерения надежности схемы. Таким образом, снизив точность измерения, можно уменьшить и вычислительную сложность построения полинома ошибки.

Использование неравномерности распределения значений сигнала на выходе

В основе предлагаемого подхода лежит тот факт, что значения сигнала на выходе большинства схем

распределены неравномерно. В частности, иногда существуют значения, которые вовсе не могут появиться на выходе схемы. Классический подход, заключающийся в использовании нескольких переключателей по большинству по одному на каждый выход схемы, может приводить к ситуации, когда на выходе всей защищенной этим методом схемы появляется запрещенное состояние, которое невозможно ни при каком входном векторе. Предлагается заменить мажорирующую часть этого подхода на схему, учитывающую неравномерность распределения значений сигнала на выходе схемы, т. е. изменить таблицу истинности мажоритарного элемента. Во всех примерах ниже будем рассматривать трехкратное мажорирование в качестве исходного подхода и попытаемся улучшить надежность, модифицируя мажоритарный элемент.

Для того чтобы дать строгое определение для оптимальных воутеров, опишем сначала обобщенный мажоритарный подход, частным случаем которого является трехкратное мажорирование.

Пусть схема S , изображенная на рис. 1, подвержена ошибкам. Значение сигнала на выходе S является разрешенным, если существует входная комбинация, приводящая к такому значению на выходе. Требуется построить такую схему V , входом которой является выход M схемы S , а результатом работы — наиболее вероятное, разрешенное значение на выходе S . Причем построенная таким образом из двух схем система должна быть эквивалентна исходной схеме, т. е. если значение на выходе S является разрешенным, то на выходе схемы V должно быть разрешенное значение. В случае когда все возможные значения сигнала на выходе S являются разрешенными, схема V в силу условия эквивалентности всегда выдает результат, полученный на входе. Таким образом, для защиты схемы S с помощью корректирующей дополнительной схемы V необходимо, чтобы у схемы S существовали запрещенные выходные значения, невозможные ни при какой входной комбинации.

Для схем, все выходные значения которых являются разрешенными, следует добавить дополни-

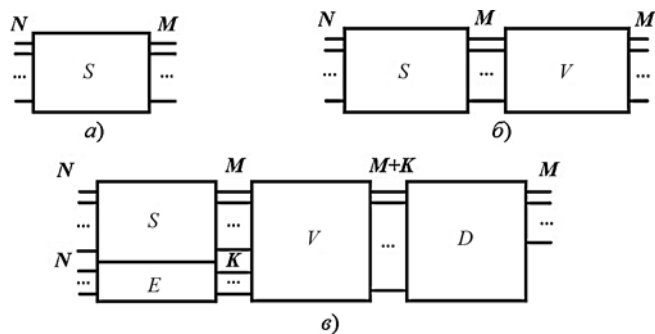


Рис. 1. Обобщенный мажоритарный подход:
a — исходная схема; *б* — схема, защищенная воутером; *в* — расширение схемы для обеспечения защиты; S — исходная схема; E — дополнительная схема; V — воутер; D — декодер

тельные выходы таким образом, чтобы у полученной расширенной схемы появились запрещенные выходные значения. Для полученной схемы можно построить нетривиальный воутер и затем отобразить полученные на выходе воутера расширенные выходные значения в значения исходной схемы.

Трехкратное мажорирование является частным случаем такого подхода. Действительно, дополнительные экземпляры схемы добавляют дополнительные выходы, создавая запрещенные выходные значения — значения, в которых не совпадают результаты экземпляров схемы. Связка воутер-декодер в данном случае представлена несколькими (по числу выходов) переключателями по большинству.

Не принимая во внимание вопрос об оптимальном расширении схемы, попытаемся для заданной схемы с запрещенными состояниями построить оптимальный воутер.

Обозначим функцию, которую выполняет схема $S: \{0, 1\}^N \times \{0, 1\}^L \rightarrow \{0, 1\}^M$, где N, M, L — число входов, выходов и элементов схемы соответственно. Будем писать $S(x, e) = y$, если на входном векторе x и векторе ошибок в элементах схемы e схема выдает результат y . Пусть теперь X и E — случайные дискретные векторные величины с распределениями

$$P(X = x) = \frac{1}{2^N} \text{ и } P(E = e) = p^{|e|}(1-p)^{L-|e|}.$$

Определим теперь функцию воутера $V(y)$. Для разрешенных значений y схемы:

$$V(y) = y$$

в силу условия эквивалентности исходной схеме.

Для запрещенных y значение воутера должно быть максимально вероятным из всех возможных, т. е.

$$V(y) = \operatorname{argmax}_z P(\{X = x, E = e | S(x, 0) = z\} | \{X = x, E = e | S(x, 0) = z \wedge S(x, e) = y\}).$$

Иными словами, значению y на входе оптимального воутера следует поставить в соответствие значение z с максимальной вероятностью того, что на выходе S был бы получен результат z , если бы она работала безошибочно, при условии, что на выходе S (с учетом возникших ошибок) получен результат y . То есть запрещенному значению y на входе оптимального воутера следует поставить в соответствие разрешенное значение z , которое имело наибольшую вероятность перехода в это запрещенное состояние.

Таким образом, оптимальный воутер задается выражением

$$V(y) = \begin{cases} y, & \text{если } \exists x S(x, 0) = y, \\ \text{иначе} \\ \operatorname{argmax}_z P(\{X = x, E = e | S(x, 0) = z\} | \{X = x, E = e | S(x, 0) = z \wedge S(x, e) = y\}). \end{cases} \quad (4)$$

Далее на конкретных примерах рассмотрим методику построения оптимальных воутеров.

Построение оптимальных воутеров

Рассмотрим сначала схему с одним выходом. В этом случае есть всего два выходных значения 0 и 1, и для схем, имеющих практическую ценность (не реализующих константу), ни одно из них не является запрещенным. Применяя трехкратное мажорирование, получим схему с тремя выходами, которые следует подать на вход мажорирующему элементу. Таблица истинности этого элемента должна удовлетворять условию эквивалентности исходной и модифицированной схем. То есть, если значения на входах мажорирующего элемента равны, он должен выдать это значение. Очевидно также, что при перестановке входов мажорирующего элемента, результат на его выходе не должен меняться. Эти условия ограничивают множество таблиц истинности мажорирующего элемента всего тремя вариантами, представленными в табл. 1.

Как будет показано в экспериментальной части, замена стандартного воутера одной из двух предложенных альтернатив в некоторых случаях приводит к значительному повышению надежности.

В случае если схема имеет несколько выходов, в классическом подходе они мажорируются независимо, с выбором результата с помощью переключателя по большинству для каждого выхода отдельно. Для некоторых схем такой подход приводит к ситуации, когда мажорирующий элемент выдает результат, заведомо невозможный для исходной схемы ни на какой входной последовательности (например, когда все экземпляры схемы выдали в качестве ре-

зультата "запрещенное" значение). Меняя таблицу истинности мажорирующего элемента, можно исключить такие ситуации, тем самым повышая отказоустойчивость конечного устройства.

Рассмотрим такое построение на примере схемы С17 с пятью входами и двумя выходами, которая представлена на рис. 2.

Рассмотрим схему S , состоящую из трех экземпляров С17 с объединенными входами.

Чтобы получить значения оптимального воутера V , построим матрицу, у которой в i -й строке и j -м столбце стоит следующая вероятность:

$$M_{ij} = P(\{X = x, E = e | S(x, 0) = i \wedge S(x, e) = j\}), \quad (5)$$

т. е. вероятность того, что при безошибочной работе схема выдаст результат i при условии, что схема выдала результат j .

Тогда, если y не является разрешенным для S , то $V(y) = \operatorname{argmax}_i M_{iy}$. Это следует из (1), если выразить условную вероятность как отношение:

$$P(B|A) = \frac{P(AB)}{P(A)}.$$

Так как в данном случае A фиксировано, можно вместо максимума условной вероятности искать максимум вероятности совместного события AB .

Матрица M_{ij} содержит в себе информацию о всех возможных элементарных исходах и сумма всех ее элементов равна единице. Выбрав какую-либо таблицу истинности для воутера, можно по этой матрице вычислить вероятность $P_{VE}(p)$ возникновения исхода, в котором воутер принимает неверное решение. Суть предлагаемого метода заключается в выборе такой таблицы истинности, для которой эта вероятность минимальна. При этом для разных p таблицы истинности оптимальных воутеров могут различаться. Чтобы избавиться от такой зависимости, будем считать p достаточно малым и для сравнения воутеров с разными таблицами истинности будем пользоваться непосредственно соответствующими полиномами P_{VE} , сравнивая их в лексикографическом порядке.

В табл. 2 представлена функция воутера, полученная описанным способом. Строки с разрешенными значениями S опущены для более компактного представления. Для комбинаций, получающихся друг из друга перестановкой, оставлен только один вариант.

На данном примере видно, что таблица истинности оптимального воутера для схемы С17 совпадает с таблицей стандартного воутера из двух переключателей по большинству. Для большого числа схем стандартный подход является оптимальным в силу незначительной неравномерности распределения выходных значений. Таблицы истинности двух воутеров для схем с более выраженной корреляцией между выходами будут различаться. Рассмот-

Таблица 1

Три варианта воутеров для схем с одним выходом

Стандартный воутер	Воутер И	Воутер ИЛИ																																																																																																												
<table border="1"> <tr><th>x_1</th><th>x_2</th><th>x_3</th><th>f</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	x_1	x_2	x_3	f	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	1	0	1	1	1	1	1	<table border="1"> <tr><th>x_1</th><th>x_2</th><th>x_3</th><th>f</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	x_1	x_2	x_3	f	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	<table border="1"> <tr><th>x_1</th><th>x_2</th><th>x_3</th><th>f</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	x_1	x_2	x_3	f	0	0	0	0	0	0	1	1	0	1	0	1	0	1	1	1	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	1
x_1	x_2	x_3	f																																																																																																											
0	0	0	0																																																																																																											
0	0	1	0																																																																																																											
0	1	0	0																																																																																																											
0	1	1	1																																																																																																											
1	0	0	0																																																																																																											
1	0	1	1																																																																																																											
1	1	0	1																																																																																																											
1	1	1	1																																																																																																											
x_1	x_2	x_3	f																																																																																																											
0	0	0	0																																																																																																											
0	0	1	0																																																																																																											
0	1	0	0																																																																																																											
0	1	1	0																																																																																																											
1	0	0	0																																																																																																											
1	0	1	0																																																																																																											
1	1	0	0																																																																																																											
1	1	1	1																																																																																																											
x_1	x_2	x_3	f																																																																																																											
0	0	0	0																																																																																																											
0	0	1	1																																																																																																											
0	1	0	1																																																																																																											
0	1	1	1																																																																																																											
1	0	0	1																																																																																																											
1	0	1	1																																																																																																											
1	1	0	1																																																																																																											
1	1	1	1																																																																																																											

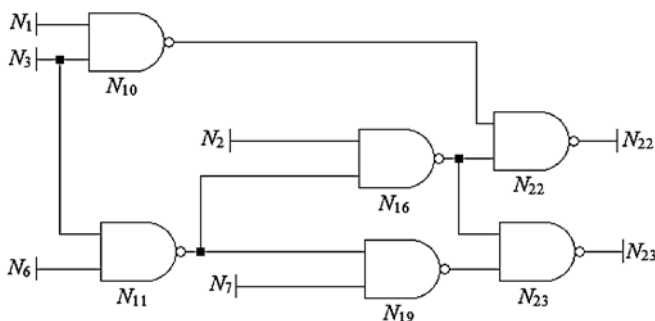


Рис. 2. Схема ISCAS85 C17

рим следующий пример. Пусть исходная схема S имеет вид, представленный на рис. 3.

Как видно, у такой схемы существует запрещенное состояние — 10. Легко показать, что существуют такие комбинации входов и ошибок на элементах, когда стандартный мажоритарный элемент

Таблица 2
Функция оптимального воутера для схемы C17

C17_1	C17_2	C17_3	$V_{optimal}$	$V_{standard}$
00	00	01	00	00
00	00	10	00	00
00	00	11	00	00
00	01	01	01	01
00	01	10	00	00
00	01	11	01	01
00	10	10	10	10
00	10	11	10	10
00	11	11	11	11
01	01	10	01	01
01	01	11	01	01
01	10	10	10	10
01	10	11	11	11
01	11	11	11	11
10	10	11	10	10
10	11	11	11	11

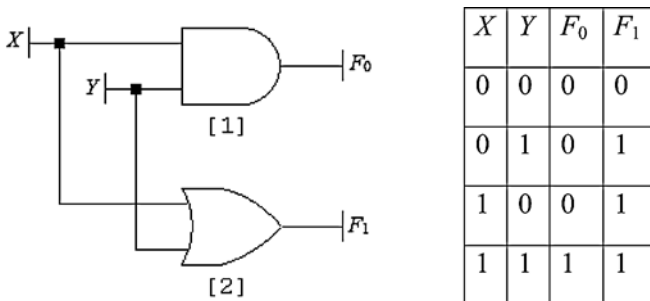


Рис. 3. Схема с запрещенным состоянием

Таблица 3
Стандартный и оптимальный воутер для схемы с запрещенным состоянием

S_1	S_2	S_3	$V_{optimal}$	$V_{standard}$
00	00	01	00	00
00	00	10	00	00
00	00	11	00	00
00	01	01	01	01
00	01	10	00	00
00	01	11	01	01
00	10	10	00	10
00	10	11	00 и 11 равновероятны	10
00	11	11	11	11
01	01	10	01	01
01	01	11	01	01
01	10	10	00 и 11 равновероятны	10
01	10	11	11	11
01	11	11	11	11
10	10	11	11	10
10	11	11	11	11

выдает заведомо неправильное запрещенное значение. Оптимальный воутер лишен этого недостатка. Таблицы истинностей для оптимального и стандартного воутеров представлены в табл. 3.

Полученная таблица истинности позволяет в рамках данного подхода достичь максимальной надежности схемы при условии безошибочности мажоритарного элемента.

В различных исследованиях, посвященных повышению отказоустойчивости, схема декодера зачастую не рассматривается в качестве объекта для сбоя и отказов [9—11]. В некоторых случаях на практике действительно стараются строить схему воутера из более надежных элементов, но чаще всего воутер также подвержен ошибкам и, так как он находится наиболее близко к выходу всей схемы, их влияние весьма велико. В нашем случае для схемы с большим числом выходов это влияние зачастую превосходит выигрыш от использования более оптимальной таблицы истинности (оптимальный воутер, как правило, сложнее стандартного и чаще ошибается). Это обстоятельство сужает границы применения такого подхода и практически не позволяет использовать его для схем с числом выходов больше двух.

Решение проблемы чрезмерного разрастания модифицированного воутера требует исследования. Одним из вариантов ее решения может служить мажорирующий элемент, который учитывает связи только нескольких, наиболее зависимых выходов, а остальные выходы мажорирует как независимые. Результаты моделирования с учетом ошибок, возникающих в воутере, рассматриваются в следующем разделе.

Результаты экспериментов

Рассмотрим некоторые схемы, неравномерность которых достаточна для получения высокого уровня отказоустойчивости при использовании нестандартных воутеров. Первая схема, изображенная на рис. 4, реализует булеву функцию $F_0(A, B, C, D, E) = ABCDE$ и принимает значение 1 только на одном наборе — (1, 1, 1, 1, 1):

Воспользовавшись подходом, описанным выше, построим для трехкратно мажорированной схемы соответствующую матрицу M вероятностей возможных исходов. По этой матрице построим зависимость $P_{VE}(p)$ для стандартного и оптимального воутеров. Вероятность ошибки воутера представлена на рис. 5.

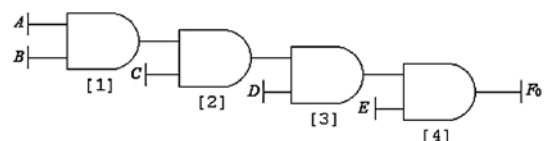


Рис. 4. Схема $F_0 = ABCDE$

Как видно, до точки $p = 0,03125$ оптимальным является стандартный воутер, однако при больших вероятностях отказа одного элемента схемы оптимальным становится воутер AND. При условии безошибочности воутера вероятность отказа всей избыточной схемы в точности равна $P_{VE}(p)$, однако для реальных воутеров, подверженных ошибкам, этот параметр является ориентировочным и не всегда отражающим реальную надежность полученной схемы. Графики зависимости вероятности ошибки схемы от вероятности отказа одного элемента для четырех рассмотренных вариантов — трехкратного мажорирования со стандартным и воутерами AND и OR, а также исходной схемы представлены на рис. 6.

Как видно из графика, воутер AND позволяет для данной схемы применить трехкратное мажорирование, тогда как стандартный воутер приводит лишь к ухудшению показателя надежности. При

этом такой прием позволяет достичь существенного уменьшения вероятности ошибки.

Проведенные эксперименты показывают, что для схем, подобных этой, а именно, реализующих булевы функции $ABCD$, $ABCDE$, $ABCDEF$, $A + B + C + D$, $A + B + C + D + E$ и т. д., замена стандартного воутера на воутер AND (или OR) дает существенный прирост надежности. Подобный результат получен также и для схем, в которых одно из выходных значений возникает достаточно редко (порядка вероятности отказа одного элемента p), а не только на одном наборе входных значений.

Рассмотрим схему, представленную на рис. 7. Она реализует булеву функцию $F_1(A, B, C, D) = ABCD + \bar{A}\bar{B}\bar{C}\bar{D}$. Результаты оценки ее отказоустойчивости, а также отказоустойчивости мажорированных схем с различными типами воутеров представлены на рис. 8.

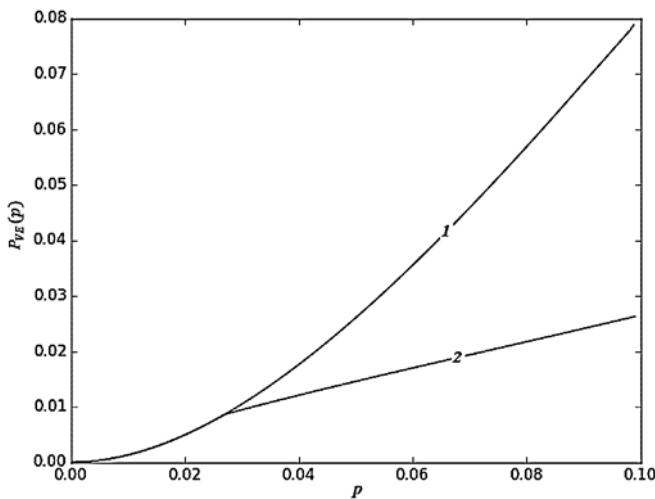


Рис. 5. Вероятность ошибки для: 1 — стандартного воутера; 2 — оптимального воутера

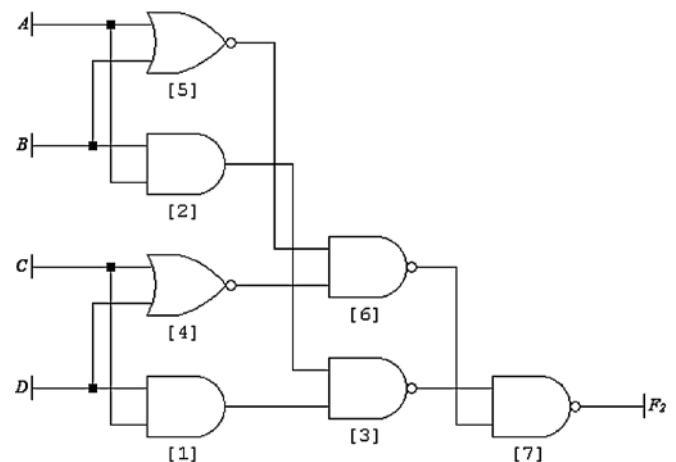


Рис. 7. Схема $F_1(A, B, C, D) = ABCD + \bar{A}\bar{B}\bar{C}\bar{D}$

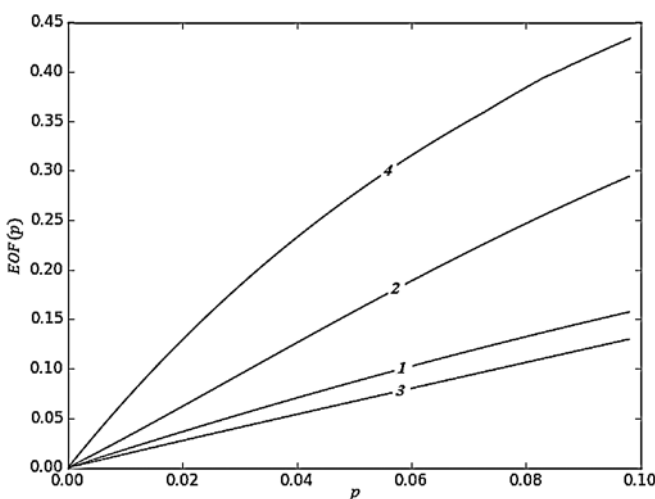


Рис. 6. $EOF(p)$ для: 1 — исходной схемы F_0 , а также F_0 , защищенной: 2 — стандартным воутером, 3 — воутером AND, 4 — воутером OR

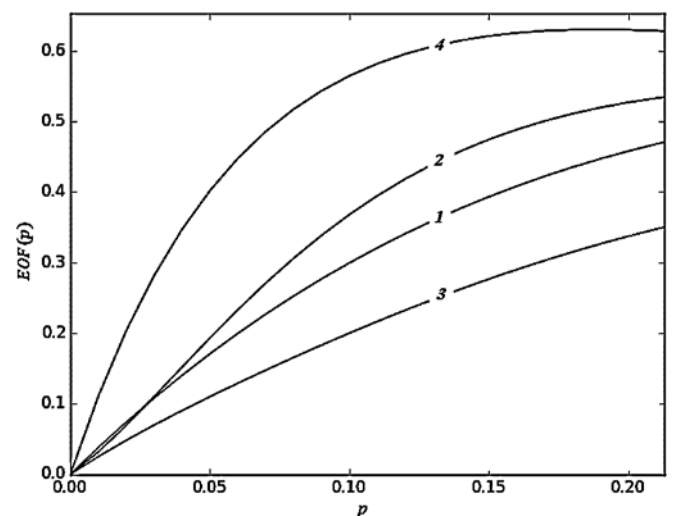


Рис. 8. $EOF(p)$ для: 1 — исходной схемы F_1 , а также F_1 , защищенной: 2 — стандартным воутером, 3 — воутером AND, 4 — воутером OR

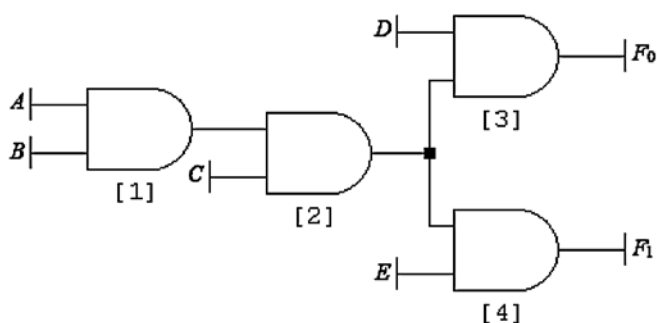


Рис. 9. Схема с двумя выходами F_2 (A, B, C, D, E)

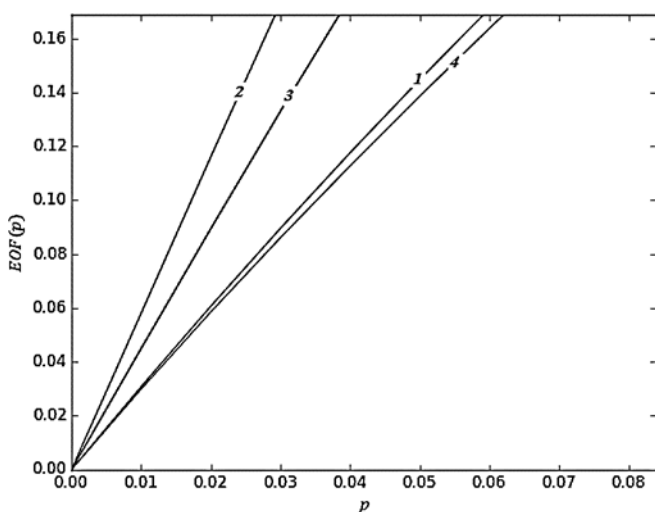


Рис. 10. $EOF(p)$ для:
1 — исходной схемы F_2 , а также F_2 , защищенной: 2 — двумя стандартными воутерами, 3 — стандартным воутером и воутером AND, 4 — двумя воутерами AND

Как видно, предлагаемая замена стандартного воутера приводит к существенному увеличению показателя надежности и для некоторых нетривиальных случаев.

Предлагаемый подход можно также применить и к схемам с числом выходов больше одного. Например, для схемы, представленной на рис. 9, были получены результаты трехкратного мажорирования с различными комбинациями из двух воутеров. Результаты в виде графиков представлены на рис. 10.

Здесь кривая 3 демонстрирует отказоустойчивость схемы, в которой первый выход был промажорирован с использованием традиционного воутера, тогда как второй выход — с помощью воутера AND. Как видно, в случае нескольких выходов, предлагаемый подход на некоторых схемах также увеличивает надежность.

Заключение

В результате проведенного исследования были получены теоретические результаты, описывающие

способ создания оптимального воутера с точки зрения выбора наиболее вероятного значения для каждой конкретной мажорируемой схемы. Это позволило обобщить подход стандартного мажорирования. Экспериментальные результаты продемонстрировали применимость данного подхода для ряда схем с высокой неравномерностью выходных данных, нетрадиционные воутеры которых получаются проще традиционных мажоритарных элементов. Для ряда схем получены их более надежные аналоги, имеющие более высокий уровень отказоустойчивости, чем схемы, полученные методом стандартного трехкратного мажорирования.

Исследование выполнено за счет гранта Российского научного фонда (проект № 14-19-01036).

Список литературы

1. Wang F., Agraval V. D. Single Event Upset: An Embedded Tutorial // Proc. of 21st International Conference on VLSI Design, Hyderabad — India, 4—8 Jan. 2008. P. 429—434.
2. Gaisler J. Evaluation of a 32-bit microprocessor with built in concurrent error-detection // Twenty-Seventh Annual International Symposium on Fault-Tolerant Computing. 1997. P. 42—46.
3. Liden P., Dahlgren P., Johansson R., Karlsson J. On Latching Probability of Particle Induced Transients in Combinational Networks // Proceedings of the 24th Symposium on Fault-Tolerant Computing (FTCS-24). 1994. P. 340—349.
4. Chen W., Gong R., Liu F., Dai K., Wang Z. Improving the Fault Tolerance of a Computer System with Space-Time Triple Modular Redundancy // Proceedings International Conference on Dependable Systems and Networks. 23—26 June 2006. P. 389—398.
5. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение // М.: Техносфера. 2006. 320 с.
6. Амербаев В. М., Корнилов А. И., Стемповский А. Л. Модулярная логарифметика — новые возможности для проектирования модулярных вычислителей и преобразователей // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). Т. 1. 2010. С. 368—373.
7. El-Maleha A. H., Oughalia F. C. A generalized modular redundancy scheme for enhancing fault tolerance of combinational circuits // Microelectronics Reliability. 2014. Vol. 54, N. 1. P. 316—326.
8. Стемповский А. Л., Тельпухов Д. В., Соловьев Р. А., Соловьев А. Н., Мячиков М. В. Моделирование возникновения неисправностей для оценки надежностных характеристик логических схем // Информационные технологии, 2014. № 11. С. 30—36.
9. Амербаев В. М., Соловьев Р. А., Тельпухов Д. В., Балака Е. С. Построение обратных преобразователей модулярной арифметики с коррекцией ошибок на базе полиадического кода // Нейрокомпьютеры: разработка, применение. 2014. № 9. С. 30—35.
10. Соловьев Р. А., Тельпухов Д. В., Балака Е. С. Устройство для вычисления скалярного произведения векторов с коррекцией ошибок на базе системы остаточных классов // Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС). 2014. № 4. С. 173—178.
11. Амербаев В. М., Балака Е. С., Щелоков А. Н. Применение структурной избыточности для повышения надежности арифметического узла вычислительного элемента бимодулярной арифметики // Известия Южного федерального университета. Технические науки. 2014. № 7 (156). С. 255—261.

A. L. Stempkovskiy, Director, D. V. Telpukhov, Head of the department, e-mail: nofrost@inbox.ru,
R. A. Solovyev, Chief Researcher, M. V. Myachikov, Student, Research Engineer IPPM RAS
Institute for Design Problems in Microelectronics

Improving the Fault Tolerance of Logic Circuits Using Unconventional Majority Voters

The article deals with the actual task of improving the reliability of the combinational microelectronic circuits. A new method of fault tolerance improvement using unconventional majority voters is proposed. Method relies on the use of non-uniformity of the signal at the output of the circuit, making it is possible to build a more effective voter. The article describes a generic approach of introducing redundancy into the scheme in order to increase reliability, and a method of constructing a truth table for optimal majority voter is also developed. In the experimental part of the paper, the proposed method was compared with the classic triple modular redundancy scheme. Particular examples show the high efficiency of the method compared to traditional approaches.

Keywords: the method of fault tolerance increase, fault tolerance, combinational circuit, triple modular redundancy

References

1. Wang F., Agraval V. D. Single Event Upset: An Embedded Tutorial. *Proc. of 21st International Conference on VLSI Design, Hyderabad — India*, 4–8 Jan., 2008, pp. 429–434.
2. Gaisler J. Evaluation of a 32-bit microprocessor with built in concurrent error-detection. *Twenty-Seventh Annual International Symposium on Fault-Tolerant Computing*, 1997, pp. 42–46.
3. Liden P., Dahlgren P., Johansson R., Karlsson J. On Latching Probability of Particle Induced Transients in Combinational Networks. *Proceedings of the 24th Symposium on Fault-Tolerant Computing (FTCS-24)*, 1994, pp. 340–349.
4. Chen W., Gong R., Liu F., Dai K., Wang Z. Improving the Fault Tolerance of a Computer System with Space-Time Triple Modular Redundancy. *Proceedings International Conference on Dependable Systems and Networks*, 23–26 June 2006, pp. 389–398.
5. Morelos-Saragosa R. *Iskusstvo pomehoustojchivogo kodirovaniya. Metody, algoritmy, primeneniye*, Moscow, Tehnosfera, 2006, 320 p.
6. Amerbaev V. M., Kornilov A. I., Stempkovskij A. L. Moduljarnaja logarifmetika — novye vozmozhnosti dlja proektirovaniya moduljarnyh vychislitelej i preobrazovatelej. *Problemy razrabotki perspektivnyh mikro- i nanojelektronnyh sistem (MES)*, 2010, vol. 1, pp. 368–373.
7. El-Maleha A. H., Oughalia F. C. A generalized modular redundancy scheme for enhancing fault tolerance of combinational circuits. *Microelectronics Reliability*, 2014, vol. 54, no. 1, pp. 316–326.
8. Stempkovskij A. L., Tel'puhov D. V., Solov'ev R. A., Solov'ev A. N., Mjachikov M. V. Modelirovanie vozniknovenija neispravnostej dlja ocenki nadezhnostnyh harakteristik logicheskikh shem. *Informacionnye tehnologii*, 2014, no. 11, pp. 30–36.
9. Amerbaev V. M., Solov'ev R. A., Tel'puhov D. V., Balaka E. S. Postroenie obratnyh preobrazovatelej moduljarnoj arifmetiki s korekciej oshibok na baze poliadicheskogo koda. *Nejrokomputery: razrabotka, primeneniye*, 2014, no. 9, pp. 30–35.
10. Solov'ev R. A., Tel'puhov D. V., Balaka E. S. Ustrojstvo dlja vychislenija skaljarnogo proizvedeniya vektorov s korekciej oshibok na baze sistemy ostatocnyh klassov. *Problemy razrabotki perspektivnyh mikro- i nanojelektronnyh sistem (MJeS)*, 2014, no. 4, pp. 173–178.
11. Amerbaev V. M., Balaka E. S., Shhelokov A. N. Primeniye strukturnoj izbytochnosti dlja povysheniya nadezhnosti arifmeticheskogo uzla vychislitel'nogo jelementa bimodul'noj arifmetiki. *Izvestija Juzhnogo federal'nogo universiteta. Tehnicheskie nauki*, 2014, no. 7 (156), pp. 255–261.

Теоретический и прикладной научно-технический журнал

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

ISSN 2220-3397

В журнале освещаются состояние и тенденции развития основных направлений индустрии программного обеспечения, связанных с проектированием, конструированием, архитектурой, обеспечением качества и сопровождением жизненного цикла программного обеспечения, а также рассматриваются достижения в области создания и эксплуатации прикладных программно-информационных систем во всех областях человеческой деятельности.

Журнал распространяется только по подписке.

Оформить подписку можно через подписные Агентства или непосредственно в редакции журнала.

Подписные индексы по каталогам:
"Роспечать" — 22765; "Пресса России" — 39795

107076, Москва, Стромынский пер., 4
Тел./факс: (499) 269-55-10
e-mail: prin@novtex.ru
<http://novtex.ru/pi.html>

ПРОГРАММНАЯ ИНЖЕНЕРИЯ SOFTWARE ENGINEERING

УДК 004.272.44, 004.94

Д. Н. Змеев, науч. сотр., e-mail: zmeievdn@ippm.ru,

А. В. Климов, ст. науч. сотр., e-mail: klimov@ippm.ru,

Н. Н. Левченко, канд. техн. наук, зав. отделом, e-mail: nick@ippm.ru,

А. С. Окунев, канд. техн. наук, вед. науч. сотр., e-mail: oku@ippm.ru,

А. Л. Стемповский, д-р техн. наук, акад. РАН, директор, e-mail: ippm@ippm.ru

Институт проблем проектирования в микроэлектронике РАН (ИППМ РАН), Зеленоград, Москва

Эмуляция аппаратно-программных средств параллельной потоковой вычислительной системы "Буран"

Описывается эмулятор, реализующий архитектуру параллельной потоковой вычислительной системы (ППВС) "Буран" на кластерных компьютерах. Эмулятор является составной частью программного комплекса для оценки возможности выполнения задач пользователя с использованием потоковой модели вычислений и определения оптимальной архитектуры. Приведены основные предпосылки к созданию программного эмулятора ППВС "Буран", особенности его построения, а также базовые алгоритмы организации основного цикла работы процесса эмуляции.

Ключевые слова: потоковая модель вычислений, эмулятор ППВС "Буран", спецвычислители, новая парадигма вычислений

Введение

Высокопроизводительные вычислительные системы кластерного типа демонстрируют низкую реальную производительность на большом круге актуальных задач и поэтому требуется изменение парадигмы вычислений [1].

Примером таких актуальных задач являются задачи из области молекулярной динамики, разнообразные графовые задачи или задачи, представляющие новое поколение прямых методов решения СЛАУ, соединяющих технику разреженных матриц с распараллеливанием вычислений, использующихся в новых САПР, работа над которыми ведется в ИППМ РАН [2, 3].

Еще более актуальна задача изменения парадигмы вычислений для разрабатываемых суперкомпьютерных систем, которые всегда были и остаются передовым фронтом в области вычислительной техники.

В связи с этим в ИППМ РАН продолжается работа над проектом параллельной потоковой вычислительной системы (ППВС) "Буран" [4], которая реализует оригинальную потоковую модель вычислений с динамически формируемым контекстом. Потоковая модель вычислений с динамически формируемым контекстом обладает рядом свойств, которые позволяют преодолевать проблемы, особенно остро проявляющиеся при повышении производительности компьютеров до эксафлопсного уровня [5, 6].

При создании вычислительной системы возникает необходимость в моделировании ее архитектуры. Поскольку архитектура разрабатываемой вычислительной системы принципиально отличается от традиционных систем, то отсутствует возможность использовать готовые компоненты для аппаратного моделирования, а из-за большого объема оборудования возникают трудности макетирования на ПЛИС.

Ранее была разработана поведенческая блочно-регистрационная модель (ПБРМ) ППВС "Буран" [7]. Данная модель — потактовая, с широким диапазоном настраиваемых параметров. На ПБРМ были исследованы задачи различных классов и получены данные [8–11], свидетельствующие о правильности выбора архитектурной реализации параллельной потоковой вычислительной системы и о высокой степени масштабирования программ, значительно превосходящей в этом показателе кластерные вычислительные системы.

Поскольку данная модель функционирует на рабочей станции, то у нее имеются ограничения на размерность пропускаемых задач и на производительность самой модели. Кроме того, эффект, связанный с увеличением степени масштабируемости, может быть получен только при моделировании тысяч вычислительных ядер системы.

В целях исследования и подтверждения полученных ранее результатов на ПБРМ в задачах большей размерности было решено создать эмулятор ППВС для работы на кластерных вычислительных системах.

Эмулятор ППВС является составной частью испытательного программного комплекса, предназначенного для оценки возможности выполнения задач пользователя с использованием потоковой модели вычислений и определения оптимальной архитектуры (с прогнозируемыми параметрами) для конкретного класса задач. Программный комплекс будет доступен пользователям через Internet и предоставит возможность создания параллельных программ и архитектур систем, адаптированных под определенный класс задач, как на базе ПЛИС, так и в виде заказного кристалла.

Потоковая модель вычислений

В основе потоковой модели вычислений с динамически формируемым контекстом лежит активация вычислительных квантов по готовности данных (рис. 1, см. третью сторону обложки). Вычислительный квант представляет собой программу, которая, будучи активированной, выполняется до конца без привлечения дополнительной информации, т. е. без приостановки процесса вычисления на подкачку дополнительных внешних данных. Различные вычислительные кванты между собой взаимодействуют и сохраняют состояние только через отправку сообщений (токенов), активирующих новые кванты. В узле-отправителе определяются и передаваемое значение, и адрес получателя, тем самым мы работаем в парадигме "раздачи" [12]. В этом состоит принципиальное отличие нашего подхода от традиционного (парадигмы "сбора"), когда вычислительный процесс сам запрашивает нужные ему данные из памяти или у других процессов.

Динамически формируемый контекст (ДФК) в нашей модели вычислений позволяет осуществлять одновременное и параллельное выполнение программы узла над различными вычислительными квантами, которые имеют разный контекст. Контекст представляет собой часть токена, по которому осуществляется поиск "совпадающих" токенов. Весь контекст полностью доступен для программиста. Программа узла получает всю необходимую информацию из контекста о местоположении токена и, соответственно, знает куда и какое число результирующих токенов послать в зависимости от выполняемых условий.

Для программирования в потоковой модели вычислений с ДФК создан параллельный язык DFL. Согласно принципам программирования параллельной потоковой вычислительной системы для программиста отсутствует представление о памяти в традиционном понимании, тем не менее, в нашей модели вычислений введено понятие виртуального адресного пространства токенов, размерность которого связана с разрядностью полей ключа токена.

Для ограниченного класса программ — так называемого линейного (или аффинного) — создается транслятор с последовательного языка программирования в параллельный язык DFL.

Архитектура параллельной потоковой вычислительной системы "Буран"

ППВС "Буран" представляет собой многоядерную масштабируемую вычислительную систему (рис. 2) [12, 13]. В состав каждого вычислительного ядра системы входят исполнительное устройство, процессор сопоставления, коммутатор токенов, блок хэширования. Между ядрами в системе передаются единицы информации в виде токенов. Токеном называется структура, в состав которой входят данное, контекст, который однозначно определяет положение операнда в виртуальном адресном пространстве задачи, и набор служебных полей. Токены являются основными информационными объектами, с которыми работает аппаратура вычислительной системы. Коммутация между ядрами осуществляется по номеру вычислительного ядра. Этот номер вырабатывается в блоке хэширования на основе содержимого полей токена и настраиваемой программистом функции распределения.

Процессор сопоставления в ППВС обеспечивает взаимодействие токенов различных типов в зависимости от содержимого поля токена "код операции", сравнивая поля ключей токенов с учетом кратности и оценивая состояние других полей токенов [14—16]. Ключи токенов сопоставляются в ассоциативной памяти ключей, а сами токены хранятся в памяти токенов.

Параллельная потоковая вычислительная система, реализующая потоковую модель вычислений с динамически формируемым контекстом, обладает рядом достоинств [17]:

- хорошая масштабируемость системы, что позволяет реализовать возможность создания как многоядерного кристалла, так и многопроцессорной вычислительной системы на базе таких кристаллов;
- аппаратное выявление неявного параллелизма задачи в ходе ее выполнения;
- наличие аппаратно-программных средств управления вычислительным процессом задачи;
- асинхронная работа отдельных блоков системы;

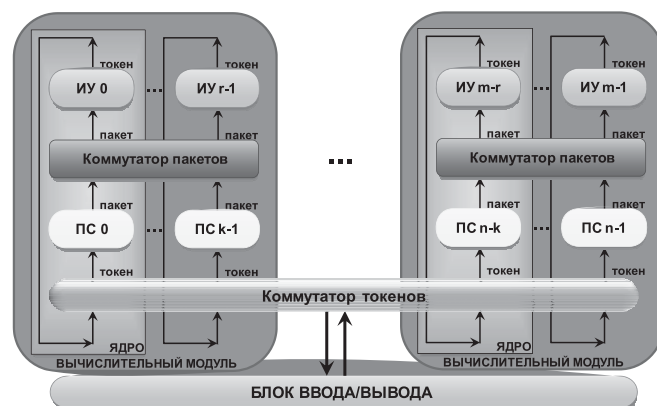


Рис. 2. Архитектура параллельной потоковой вычислительной системы "Буран"

- нивелирование задержки в коммуникационной сети за счет потоковой организации вычислительного процесса и др.

Необходимость создания эмулятора

Создание программного эмулятора параллельной потоковой вычислительной системы "Буря" преследует несколько целей.

Во-первых, эмулятор представляет собой часть аппаратно-программного комплекса (АПК). В состав АПК входят: средства параллельного языка DFL, потактовая модель ПБРМ, эмулятор ППВС, RTL-модель вычислительного ядра и вычислительного модуля, макет.

Во-вторых, эмулятор позволяет рядовым пользователям программировать в парадигме "раздачи", исследовать прохождение параллельных программ в привычной для них среде, используя при этом модель вычислений с управлением потоком данных. Также можно будет апробировать разработанные средства "аппаратного" масштабирования программ и оценить потенциал потоковой модели вычислений.

В-третьих, разработчики вычислительной системы "Буря" используют эмулятор для отработки различных вариантов аппаратных решений и получения статистической информации для оценки эффективности их внедрения.

В-четвертых, эмулятор используется для исследования работы ППВС "Буря" на большом числе вычислительных ядер для широкого круга задач с существенно большей их размерностью по сравнению с поведенческой блочно-регистрационной моделью.

В-пятых, поскольку создаваемые в нашей парадигме программы обычно обладают хорошей масштабируемостью, то несмотря на относительно высокие накладные расходы (на сам процесс эмуляции), при распараллеливании на большое число вычислительных ядер эмулятор может демонстрировать вполне конкурентную эффективность по сравнению с кластерными вычислительными системами, что для некоторых задач позволит снизить время их решения.

Ускорения работы на программном эмуляторе ППВС удалось добиться как за счет параллельной реализации модели системы, так и за счет упрощения модели, отказавшись от потактового моделирования. При этом программный эмулятор ППВС обеспечивает максимально возможную производительность, сохраняя при этом функциональность системы. Потактовое моделирование вносит синхронность в процессы, являющиеся имманентно асинхронными, что в результате препятствует распараллеливанию и значительному замедлению из-за простоев в синхронизации и ожидании обменов. Было решено отказаться от потактового моделирования еще и потому, что с точки зрения оценки адекватности моделирования отсутствие учета времени (тактов) не будет являться критичным. Для оценки адекватности моделирования существуют

такие оценки работы, как число выполненных узлов, число порожденных токенов, число токенов, переданных по сети (реальной или имитируемой). Подсчет необходимых показателей можно вести опционально, указывая необходимые непосредственно перед запуском процесса моделирования программы.

Описание эмулятора ППВС

Эмулятор ППВС является статической библиотекой на языке С с использованием интерфейса обмена сообщениями (MPI) и предназначен для работы на UNIX-системах.

Библиотеки и функции. В видимой части библиотеки доступны функции инициализации эмулятора и формирования токенов, а также параметры, через которые программист может влиять на процесс. В закрытой части реализован алгоритм сопоставления токенов и процесс их приема-передачи между вычислительными ядрами.

При разработке программы с использованием библиотеки эмулятора ППВС программист перестраивает алгоритм последовательной программы в парадигму "раздачи" [12], в которой функции не называются явно, а формируются токены с указателями на них. Каждая такая функция должна быть определенного типа. При формировании токена, например, следует пользоваться функцией DFE_SendEx или одним из ее переопределений: DFE_Send — для простого токена; DFE_MSend — для многоходового токена; DFE_GSend — для глобального токена; DFE_HSend — для выходного токена.

Запуск эмулятора ППВС. Инициализация и запуск эмулятора проводятся передачей эмулятору указателей на все узлы с помощью соответствующей функции. Это делается для того, чтобы эмулятор знал, какой узел вызывать при совпадении ключей токенов в памяти сопоставления. Далее в любом порядке устанавливаются параметры эмулятора с помощью функции формирования начальных данных и функции, учитывающей метод формирования начальных данных. Существует два варианта метода формирования токенов (начальных данных задачи):

- все токены формируются в нулевом ядре и затем рассылаются на другие ядра в соответствии с функцией распределения вычислений;
- токены формируются только в ядре, которому они "принадлежат" в соответствии с функцией распределения вычислений.

В эмуляторе реализованы три базовых функции распределения вычислений: "стандартная", "по полю" и "смещение полей". Также можно указать и свою функцию распределения вычислений, при этом единственным требованием является то, чтобы результат вычисления номера ядра был в пределах количества выделенных ядер.

Определение окончания эмуляции. Для определения окончания процесса эмуляции реализовано два способа. Первый способ — "по числу выходных токенов". В этом случае все выходные токены акку-

мулируются в нулевом ядре, и когда их число достигает установленного значения, на все ядра посылается сигнал прекращения эмуляции. Вторым способом — "определение тишины". Он реализован следующим образом: когда у нулевого ядра на входе нет токенов для обработки, оно рассылает остальным ядрам запрос об их состоянии. В качестве ответа нулевое ядро получает разность значений между полученными и отправленными токенами каждого вычислительного ядра. Если сумма этих "разностей" для всех ядер равна нулю, то это означает, что в эмуляторе не осталось необработанных токенов. Для "гарантированного" определения окончания проводится контрольный "запрос состояний ядер" и в случае повторения результата рассылается сигнал о прекращении эмуляции.

Работа эмулятора ППВС

Рабочий цикл эмулятора можно условно разбить на три этапа: инициализация, эмуляция и завершение работы.

Инициализация. На этапе инициализации эмулятора происходит:

- выделение памяти под буфера и области хранения токенов;
- заполнение таблицы сопоставления;
- исполнение пользовательской функции преобразования начальных данных в токены.

Эмуляция. Этап собственно эмуляции разделен на две части: ожидание токенов и обработка токенов. Токены, сформированные из начальных данных, поступают на вход эмулятора. Все пришедшие в ядро токены хранятся во входном буфере, реализованном по принципу FIFO. Далее из этого буфера извлекается первый токен и анализируется тип токена. Если данный токен является сигналом прекращения эмуляции, то функция незамедлительно переходит к этапу завершения работы. Если токен является выходным, то он записывается в список выходных токенов. Во всех остальных случаях токен обрабатывается функцией сопоставления.

Пришедший токен в процессе сопоставления последовательно сравнивается со всеми записанными ранее в память токенами. Сначала сравниваются их контексты с учетом комбинированной маски. При положительном результате сравниваются коды команд (в соответствии с таблицей сопоставления) и типы взаимодействия. Если оба сравнения прошли удачно, то в зависимости от кода команд выполняются те или иные действия над обоими токенами. Результатом этих действий может быть:

- удаление токена из памяти;
- формирование пакета;
- изменение токена;
- новая посылка пришедшего токена на обработку.

При формировании пакета определяется номер узла для его обработки, после чего он вызывается с указателем на пакет в качестве параметра. При выполнении узла формируются новые токены. Для

каждого такого токена с помощью функции распределения рассчитывается номер ядра. Если этот номер совпадает с текущим ядром, токен помещается во входной буфер. В противном случае он помещается в выходной буфер (свой для каждого ядра). При переполнении выходного буфера его содержимое с помощью функции MPI_Send передается в соответствующее ядро, не дожидаясь обработки оставшихся токенов, после чего буфер обнуляется. По завершении выполнения узла пришедший токен записывается в память только в том случае, если указатель на него не обнулен, а обнуление происходит при исчерпании токеном кратности.

По завершении обработки всех входных токенов, а она продолжается, пока входной буфер не опустеет, проводится контрольная рассылка содержимого выходных буферов на соответствующие ядра, после чего функция возвращается на этап ожидания токенов.

Ожидание токенов происходит при использовании блокирующей функции MPI_Probe, выход из которой означает получение списка токенов или "запроса состояний ядер". Для нулевого ядра отличие заключается в том, что "прослушивание состояния ядер" проводится неблокирующей функцией MPI_Iprobe, результатом выполнения которой является информация о наличии или отсутствии на входе каких-либо сигналов.

При получении новых токенов они последовательно переносятся из списка во входной буфер, после чего эмулятор переходит к этапу их обработки.

Завершение работы. Этап завершения работы включает в себя обработку выходных токенов, вывод на терминал статистических показателей и освобождение выделенной в процессе эмуляции памяти.

Собираемая статистика. В качестве статистических показателей на терминал выводятся следующие данные:

- общее время эмуляции;
- время формирования входных токенов;
- время выдачи токенов на HOST;
- время сопоставления;
- время выполнения узлов;
- число сопоставлений токенов;
- объем переданных данных в байтах;
- число переданных токенов.

Все временные показатели вычисляются как среднее значение для всех ядер, а количественные показатели суммируются.

Работа эмулятора на программах

Для проверки работы эмулятора ППВС выбирались задачи, которые требуют больших вычислительных ресурсов.

Традиционно для решения этих задач нужны высокомасштабируемые программы, однако большинство программ, выполняемых на высокопроизводительных системах, создаются с использованием "простого" параллелизма (разбиваются на части) и плохо масштабируются.

Это касается, например, задач молекулярной динамики. Для них существует много алгоритмов, которые пытаются распараллеливать на кластерных системах. Причем для этих алгоритмов (от "простых" в реализации, до более "трудных" классов — алгоритмов "пространственной декомпозиции") распараллеливание по данным оказалось проблематичным из-за нерегулярности этого класса задач. Алгоритм NAMD2 [18], например, написан на Charm++ и доказал свою способность распараллеливаться на тысячу ядер, однако этот алгоритм не может быть распараллелен на десятки и сотни тысяч ядер. И, вообще, известные алгоритмы для задач этой области трудно распараллеливались, так как построены на синхронизации после каждого шага.

Эксперименты с созданным эмулятором ППВС проводились на высокопроизводительных кластерах МСЦ РАН, на кластерах "Чебышев" и "Ломоносов" в МГУ. Необходимо отметить, что для прогона задач на эмуляторе ППВС DFL-код программы транслируется на язык C и сопрягается с библиотекой, реализуемой с использованием MPI.

Вместе с программой пользователем задается некая функция распределения вычислений (формула, зависящая от полей контекста). При выборе функции следует обеспечить относительную равномерность загрузки ядер и минимизировать число посылок токенов между вычислительными ядрами.

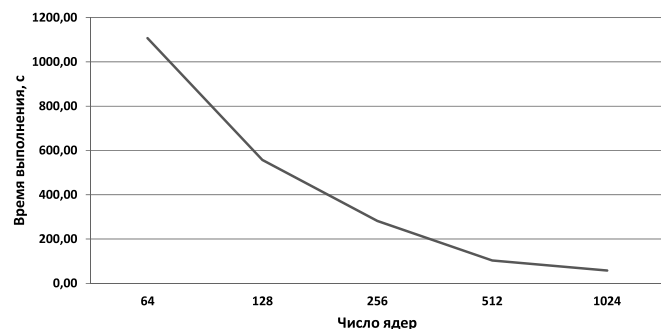


Рис. 3. Зависимость времени выполнения программы "Теплопроводность" от числа ядер

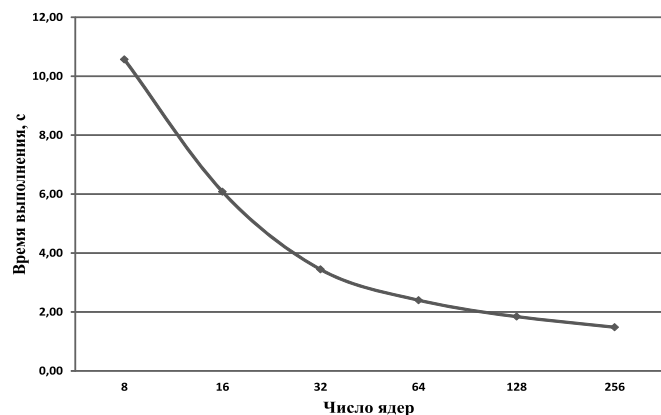


Рис. 4. Зависимость времени выполнения задачи "Молекулярная динамика" от числа ядер на суперкомпьютере "Ломоносов", 64к частиц, 100 итераций, 128 кубоидов

На ряде задач получены хорошие результаты с точки зрения повышения степени масштабируемости программ. В частности, на кластерах МГУ удалось промоделировать прохождение задач на тысячах ядер, чего не удавалось ранее. Полученные результаты анализируются. Программируются и готовятся к запуску новые актуальные задачи.

В качестве примеров приведены результаты прогона задачи "Теплопроводность" на эмуляторе (кластер "Чебышев") (рис. 3) и задачи из области молекулярной динамики [19] (суперкомпьютер "Ломоносов") (рис. 4), демонстрирующие хороший рост производительности в зависимости от увеличения числа ядер.

Для задачи из области молекулярной динамики сначала были проведены экспериментальные расчеты на потактовой регистровой модели ППВС. В силу ограничений, накладываемых этой моделью, использовали только задачи объемом 64 частицы, 256 частиц и 1024 частицы. Затем созданная программа была адаптирована (со значительным увеличением расчетных данных) для ее прохождения на эмуляторе вычислительной системы на суперкомпьютерах кластерного типа, таких как "Ломоносов" и "Чебышев", установленных в МГУ.

Эксперименты проводили на различных конфигурациях системы — от 8 до 256 вычислительных ядер. На рис. 4 видно, что с ростом числа ядер происходит практически линейное уменьшение времени выполнения задачи на эмуляторе. Это означает, что размерность самой задачи и параллелизм, заложенный в реализованном алгоритме, обеспечивают эффективное использование имеющихся вычислительных ресурсов.

На эмуляторе нашей системы на задаче с числом атомов 16К (т. е. меньшим в два раза) и даже на задаче размерностью 4К масштабирование продолжается вплоть до 256 ядер. Надо отметить, что часть вычислительной нагрузки при работе эмулятора составляют накладные расходы на организацию работы самого эмулятора.

Вообще ускорение работы средств моделирования на эмуляторе составило более трех порядков по сравнению с моделированием работы ППВС на программной блочно-регистровой модели системы, так как регистровая модель работает на обычном персональном компьютере.

Заключение

В статье описаны принципы и особенности организации эмуляции архитектуры ППВС "Буря" на кластерных компьютерах, базовые функции эмулятора и основные режимы его работы.

На основе проведенных работ уже сейчас можно сделать вывод о том, что наличие такого эмулятора ППВС и модели вычислений с управлением потоком данных может рассматриваться как альтернативная модель параллельного программирования для традиционных кластерных суперкомпьютеров.

Создаваемые в ней программы обычно обладают высокой степенью масштабируемости и потому, несмотря на относительно высокие накладные расходы, при большом числе MPI-процессов могут продемонстрировать вполне конкурентную эффективность.

Помимо этого благодаря разработке эмулятора создается также альтернативное направление для создания спецвычислителей в новой парадигме вычислений. Эмулятор позволит разработчикам оценить перспективность создания той или иной архитектуры, адаптированной под конкретную задачу в новой парадигме вычислений.

Эмулятор позволяет также перейти к макетированию, а так как в ИППМ РАН сейчас ведутся работы, связанные с оптимальным выбором состава библиотечных элементов (RTL-синтез) для создания отечественных ПЛИС [20], возможен вариант реализации макета нашей системы на таких ПЛИС.

Программистам эмулятор предоставит возможность программировать в парадигме "раздачи", пропускать эти программы, используя модель вычислений с управлением потоком данных, апробировать разработанные средства "аппаратного" масштабирования программ, а в будущем эмулятор станет частью программного комплекса определения параметров оптимальной архитектуры спецвычислителей для задач пользователей.

Разработчикам вычислительной системы эмулятор позволит обрабатывать различные варианты аппаратных решений, получать статистическую информацию и оценку эффективности от их внедрения. В перспективе планируется промоделировать с помощью эмулятора задачи, использующие модулярные алгоритмы для создания отказоустойчивых спецвычислителей [21].

Работа выполнена с использованием ресурсов суперкомпьютерного комплекса МГУ имени М. В. Ломоносова [22].

Список литературы

1. **Стемпковский А. Л., Климов А. В., Левченко Н. Н., Окунев А. С.** Методы адаптации параллельной потоковой вычислительной системы под задачи отдельных классов // Информационные технологии и вычислительные системы. 2009. № 3. С. 12–21.
2. **Гуарий М. М., Жаров М. М., Русаков С. Г., Ульянов С. Л.** Методы моделирования в частотной области периодических режимов автогенераторных схем // Информационные технологии. 2014. № 10. С. 36–43.
3. **Гуарий М. М., Жаров М. М., Русаков С. Г., Ульянов С. Л.** Методы моделирования в частотной области нелинейных радиотехнических схем // Вестник РГРТУ. 2015. № 1 (выпуск 51). С. 52–60.
4. **Донгарра Дж.** Свежий взгляд из-за океана // Суперкомпьютеры. 2013. № 2 (14). С. 6–8.
5. **Горбунов В., Эйсымонт Л.** Экзафлопсный барьер: проблемы и решения // Открытые системы. 2010. № 5. С. 12–15.
6. **Горбунов В., Елизаров Г., Эйсымонт Л.** Экзафлопсные суперкомпьютеры: достижения и перспективы // Открытые системы. 2013. № 97. С. 10–14.
7. **Левченко Н. Н., Окунев А. С., Змеев Д. Н.** Расширение возможностей поведенческой блочно-регистровой модели параллельной потоковой вычислительной системы // Матер. Пятой Международной молодежной научно-технической конференции "Высокопроизводительные вычислительные системы (ВПВС 2008)". Таганрог, Россия, 2008. С. 371–374.

8. **Левченко Н. Н., Окунев А. С.** Исследование применения многоходовых узлов в параллельной потоковой вычислительной системе для задач фильтрации // Матер. Пятой Международной молодежной научно-технической конференции "Высокопроизводительные вычислительные системы (ВПВС 2008)". Таганрог, Россия, 2008. С. 367–370.

9. **Левченко Н. Н., Окунев А. С., Климов А. В.** Реализация задачи "Повышение контрастности изображения" на параллельной потоковой вычислительной системе // Матер. Международной научно-технической конференции "Многопроцессорные вычислительные и управляющие системы (МВУС—2009)". Дивноморское, Россия, ТТИ ЮФУ, 2009. Т. 1. С. 197–200.

10. **Левченко Н. Н., Окунев А. С.** Анализ прохождения задачи "обнаружение дефектов" на параллельной потоковой вычислительной системе // Труды Третьей Всероссийской научной конференции "Методы и средства обработки информации". М.: МАКС Пресс, 2009. С. 346–350.

11. **Андреев П. П., Левченко Н. Н., Окунев А. С., Цветков В. В.** Реализация алгоритма сжатия изображения jpeg на параллельной потоковой вычислительной системе // Матер. Международной научно-технической конференции "Многопроцессорные вычислительные и управляющие системы (МВУС—2009)". Дивноморское, Россия, ТТИ ЮФУ, 2009. Т. 1. С. 126–128.

12. **Климов Арк. В., Окунев А. С.** Потоковая модель вычислений как путь к экзафлопсу // Электрон. изд.: Тр. междунар. суперкомпьютерной конференции "Научный сервис в сети ИНТЕРНЕТ: экзафлопсное будущее", Новороссийск, 19–24 сентября 2011 г., [Электронный ресурс]. М.: Изд-во МГУ, 2011. С. 261–266. URL: <http://agora.guru.ru/abrau2011/pdf/261.pdf> (дата обращения: 05.05.2015).

13. **Климов Арк. В., Левченко Н. Н., Окунев А. С.** Модель вычислений с управлением потоком данных как средство решения проблем больших распределенных систем // Матер. Второй Всероссийской научно-технической конференции "Суперкомпьютерные технологии" (СКТ-2012). Дивноморское, Геленджикский район, 24–29 сентября 2012 года. 2012.

14. **Левченко Н. Н., Окунев А. С., Яхонтов Д. Е.** Исследование работы процессора сопоставления параллельной потоковой вычислительной системы "Буран" // Всероссийская научно-техническая конф. "Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС — 2012): сб. тр. М.: ИППМ РАН, 2012. С. 467–470.

15. **Левченко Н. Н., Окунев А. С., Шурчков И. О., Яхонтов Д. Е.** Варианты реализации контроллера памяти параллельной потоковой вычислительной системы для работы с векторными и многоходовыми узлами // Всероссийская научно-техническая конф. "Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС — 2012): сб. тр. М.: ИППМ РАН, 2012. С. 463–466.

16. **Змеев Д. Н., Левченко Н. Н., Окунев А. С., Климов А. В.** Архитектура планировщика процессора сопоставления ППВС "Буран" // Всероссийская научно-техн. конф. "Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС): сб. трудов. М.: ИППМ РАН, 2014. № 4. С. 75–78.

17. **Климов Арк. В., Левченко Н. Н., Окунев А. С.** Преимущества потоковой модели вычислений в условиях неоднородных сетей // Информационные технологии и вычислительные системы. 2012. № 2. С. 36–45.

18. **James C. Phillips, Rosemary Braun et al.** Scalable molecular dynamics with NAMD // Journal of Computational Chemistry. 2005. Vol. 26. P. 1781–1802.

19. **Климов А. В., Левченко Н. Н., Окунев А. С., Змеев Д. Н., Стемпковский А. Л.** Исследование возможности асинхронной реализации задачи молекулярной динамики на ППВС "Буран" // Качество. Инновации. Образование. 2014. № 10. С. 46–51.

20. **Гаврилов С. В., Глебов А. Л., Стемпковский А. Л.** Структурная оптимизация цифровых КМОП схем // Информационные технологии и вычислительные системы. 2002. № 4. С. 40–47.

21. **Стемпковский А. Л., Тельпухов Д. В., Соловьев Р. А., Соловьев А. Н., Мячиков М. В.** Моделирование возникновения неисправностей для оценки надежностных характеристик логических схем // Информационные технологии. 2014. № 11. С. 30–36.

22. **Воеводин Вл. В., Жуматий С. А., Соболев С. И., Антонов А. С., Брызгалов П. А., Никитенко Д. А., Стефанов К. С., Воеводин Вад. В.** Практика суперкомпьютера "Ломоносов" // Открытые системы. 2012. № 7. С. 36–39.

D. N. Zmejev, Researcher, e-mail: zmejevdn@ippm.ru,
A. V. Klimov, Senior Researcher, e-mail: klimov@ippm.ru,
N. N. Levchenko, Ph. D., Head of Department, e-mail: nick@ippm.ru,
A. S. Okunev, Ph. D., Leading Researcher, e-mail: oku@ippm.ru,
A. L. Stempkovsky, Academician, Director of IDPM RAS, e-mail: ippm@ippm.ru
Institute for Design Problems in Microelectronics of Russian Academy of Sciences

Emulation on Hardware and Software of the Parallel Computing Streaming System "Buran"

The article describes the emulator, that implements the architecture of parallel dataflow computing system (PDCS) "Buran" on cluster computers. The emulator is a part of the software complex to assess the possibility of tasks execution by using a parallel dataflow computing system and to determine the optimal architecture. The article describes the main preconditions for the creation of a software emulator of PDCS "Buran", construction features, as well as basic algorithms of the organization of the main cycle of the emulation process.

Keywords: dataflow computation model, emulator of PDCS "Buran", special calculator, new paradigm of computing

References

1. Stempkovskij A. L., Klimov A. V., Levchenko N. N., Okunev A. S. Metody adaptacii parallel'noj potokovoj vychislitel'noj sistemy pod zadachi otdel'nyh klassov. *Informacionnye tehnologii i vychislitel'nye sistemy*, 2009, no. 3, pp. 12–21 (in Russian).
2. Gurarij M. M., Zharov M. M., Rusakov S. G., Ul'janov S. L. Metody modelirovaniya v chastotnoj oblasti periodicheskikh rezhimov avtogeneratornyh shem. *Informacionnye tehnologii*, 2014, no. 10, pp. 36–43 (in Russian).
3. Gurarij M. M., Zharov M. M., Rusakov S. G., Ul'janov S. L. Metody modelirovaniya v chastotnoj oblasti nelinejnyh radiotekhnicheskikh shem. *Vestnik RGRU*, 2015, no. 1, pp. 52–60 (in Russian).
4. Dongarra Dzh. Svezhij vzgljad iz-za okeana. *Superkomp'yutery*. 2013, no. 2 (14), pp. 6–8 (in Russian).
5. Gorbunov V., Jejsymont L. Ekzaflopsnyj bar'er: problemy i reshenija. *Otkrytie sistemy*. 2010, no. 5, pp. 12–15 (in Russian).
6. Gorbunov V., Elizarov G., Jejsymont L. Jekzaflopsnye superkomp'yutery: dostizhenija i perspektivy. *Otkrytie sistemy*, 2013, no. 07, pp. 10–14 (in Russian).
7. Levchenko N. N., Okunev A. S., Zmeev D. N. Rasshirenie vozmozhnostej povedencheskoj blochno-registrovoj modeli parallel'noj potokovoj vychislitel'noj sistemy. *Materialy Pjatoj Mezhdunarodnoj molodezhnoj nauchno-tehnicheskaj konferencii "Vysokoproizvoditel'nye vychislitel'nye sistemy (VPVS 2008)"*, Taganrog, Russia, 2008, pp. 371–374 (in Russian).
8. Levchenko N. N., Okunev A. S. Issledovanie primeneniya mnogovhodovyh uzlov v parallel'noj potokovoj vychislitel'noj sisteme dlja zadach fil'tracii. *Materialy Pjatoj Mezhdunarodnoj molodezhnoj nauchno-tehnicheskaj konferencii "Vysokoproizvoditel'nye vychislitel'nye sistemy (VPVS 2008)"*, Taganrog, Russia, 2008, pp. 367–370 (in Russian).
9. Levchenko N. N., Okunev A. S., Klimov A. V. Realizacija zadachi "Povyshenie kontrastnosti izobrazhenija" na parallel'noj potokovoj vychislitel'noj sisteme. *Materialy Mezhdunarodnoj nauchno-tehnicheskaj konferencii "Mnogoprocessornye vychislitel'nye i upravljajushhie sistemy (MVUS-2009)"*, Divnomorsk, Russia, 2009, vol. 1, pp. 197–200 (in Russian).
10. Levchenko N. N., Okunev A. S. Analiz prohozhdenija zadachi "obnaruzhenie defektov" na parallel'noj potokovoj vychislitel'noj sisteme. *Trudy Tret'ej Vserossijskoj nauchnoj konferencii "Metody i sredstva obrabotki informacii"*, Moscow, 2009, pp. 346–350 (in Russian).
11. Andreev P. P., Levchenko N. N., Okunev A. S., Cvetkov V. V. Realizacija algoritma szhatija izobrazhenija jpeg na parallel'noj potokovoj vychislitel'noj sisteme. *Materialy Mezhdunarodnoj nauchno-tehnicheskaj konferencii "Mnogoprocessornye vychislitel'nye i upravljajushhie sistemy (MVUS-2009)"*, Divnomorsk, Russia, 2009, vol. 1, pp. 126–128 (in Russian).
12. Klimov Ark. V., Okunev A. S. Jelektron. izd.: Potokovaja model' vychislenij kak put' k jekzaflopsu. *Trudy mezhdunarodnoj superkomp'yuternoj konferencii "Nauchnyj servis v seti INTERNET: jekzaflopsnoe budushhee"*, 2011, URL: <http://agora.guru.ru/abrau2011/pdf/261.pdf> (accessed: 05.05.2015) (in Russian).
13. Klimov Ark. V., Levchenko N. N., Okunev A. S. Model' vychislenij s upravleniem potokom dannyh kak sredstvo reshenija problem bol'shix raspredelennyh system. *Materialy Vtoroj Vserossijskoj nauchno-tehnicheskaj konferencii "Superkomp'yuternye tehnologii" (SKT-2012)*, Divnomorsk, Russia, 2012 (in Russian).
14. Levchenko N. N., Okunev A. S., Jahontov D. E. Issledovanie raboty processora sopostavlenija parallel'noj potokovoj vychislitel'noj sistemy "Buran". *Vserossijskaja nauchno-tehnicheskaja konferencija "Problemy razrabotki perspektivnyh mikro- i nanojelektronnyh sistem (MJeS - 2012)"*: sbornik trudov, 2012, pp. 467–470 (in Russian).
15. Levchenko N. N., Okunev A. S., Shurchkov I. O., Jahontov D. E. Varianty realizacii kontrollera pamjati parallel'noj potokovoj vychislitel'noj sistemy dlja raboty s vektornymi i mnogovhodovymi uzlamy. *Vserossijskaja nauchno-tehnicheskaja konferencija "Problemy razrabotki perspektivnyh mikro- i nanojelektronnyh sistem (MJeS - 2012)"*: sbornik trudov, 2012, pp. 463–466 (in Russian).
16. Zmeev D. N., Levchenko N. N., Okunev A. S., Klimov A. V. Arhitektura planirovshhika processora sopostavlenija PPVS "Buran". *Vserossijskaja nauchno-tehnicheskaja konferencija "Problemy razrabotki perspektivnyh mikro- i nanojelektronnyh sistem (MJeS)"*: sbornik trudov, 2014, no. 4, pp. 75–78 (in Russian).
17. Klimov Ark. V., Levchenko N. N., Okunev A. S. Preimushhestva potokovoj modeli vychislenij v uslovijah neodnorodnyh setej. *Informacionnye tehnologii i vychislitel'nye sistemy*, 2012, no. 2, pp. 36–45 (in Russian).
18. James C. Phillips, Rosemary Braun et al. Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry*, 2005, vol. 26, pp. 1781–1802.
19. Klimov A. V., Levchenko N. N., Okunev A. S., Zmeev D. N., Stempkovskij A. L. Issledovanie vozmozhnosti asinhronnoj realizacii zadachi molekulyarnoj dinamiki na PPVS "Buran". *Kachestvo. Innovacii. Obrazovanie*, 2014, no. 10, pp. 46–51 (in Russian).
20. Gavrilov S. V., Glebov A. L., Stempkovskij A. L. Strukturnaja optimizacija cifrovych KMOP shem. *Informacionnye tehnologii i vychislitel'nye sistemy*, 2002, no. 4, pp. 40–47 (in Russian).
21. Stempkovskij A. L., Tel'puhov D. V., Solov'ev R. A., Solov'ev A. N., Mjachikov M. V. Modelirovanie vzniknovenija neispravnostej dlja ocenki nadezhnostnyh harakteristik logicheskikh shem. *Informacionnye tehnologii*, 2014, no. 11, pp. 30–36 (in Russian).
22. Voevodin V. V., Zhumatij S. A., Sobolev S. I., Antonov A. S., Bryzgalov P. A., Nikitenko D. A., Stefanov K. S., Voevodin Vad. V. Praktika superkomp'yutera "Lomonosov". *Otkrytie sistemy*, 2012, no. 7, pp. 36–39 (in Russian).

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОРГАНИЗАЦИОННЫХ И СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ INFORMATION TECHNOLOGIES IN THE ORGANIZATIONAL AND SOCIO-ECONOMIC SYSTEMS

УДК 519.86

С. И. Спивак¹, д-р физ.-мат. наук, проф., зав. кафедрой, semen.spivak@mail.ru,
О. Г. Кантор², канд. физ.-мат. наук, доц., ст. науч. сотрудник, e-mail: o_kantor@mail.ru

¹ ФГБОУ ВПО Башкирский государственный университет

² ФГБУН Институт социально-экономических исследований Уфимского научного центра РАН

Чебышевское приближение в задачах эконометрического моделирования¹

Исследована возможность применения идеи задачи чебышевского приближения к проблеме идентификации эконометрических моделей. Показана связь оптимального решения задачи чебышевского приближения со средней ошибкой аппроксимации — критерием точности модели. Предложенные подходы к оценке параметров эконометрических моделей нивелируют ряд объективно существующих проблем и позволяют учитывать важные с точки зрения исследователя требования непосредственно на этапе идентификации.

Ключевые слова: идентификация модели, верификация модели, чебышевское приближение

Введение

В качестве основного инструментария для эконометрического моделирования используются математико-статистические методы [1]. Их применение осуществляется в соответствии с шестью основными этапами, в рамках которых проводятся, соответственно, постановка проблемы, предмодельный анализ, параметризация, сбор необходимой информации, идентификация модели и ее верификация. К числу основных проблем при этом относятся спецификация модели (т. е. определение непосредственного вида модели), ее идентификация — проблема, связанная с выбором и непосредственной реализацией методов оценивания параметров модели, и верификация — проверка соответствия полученной модели ряду характеристик, обуславливающих ее качественные признаки, такие как, например, точность и адекватность [1].

Важное значение имеют субъективные характеристики исследователя. Так, спецификация модели во многом зависит от того, насколько хорошо ис-

следователь разбирается в сути проблемы и, безусловно, от его профессионализма и компетенции. Однако помимо субъективных существуют и некоторые объективные обстоятельства, оказывающие влияние на процесс и результаты эконометрического моделирования. В их числе можно отметить, как минимум, следующие два. Во-первых, непосредственное определение параметров модели (ее идентификация) осуществляется с учетом имеющейся экспериментальной информации, состав и качество которой во многом определяют возможности применения тех или методов. Так, например, известно, что для успешного применения математико-статистического инструментария число наблюдений (экспериментальных данных) должно в 6—7 раз превышать число параметров при экзогенных факторах [2]. Однако далеко не во всех практических исследованиях такое количество данных может быть обеспечено. Во-вторых, в процессе верификации необходимо проверять ограничивающие условия на значения параметров и (или) переменных модели, которые, очевидно, должны быть формализованы в виде каких-либо математических выражений. Классические математико-статистические методы, применяемые в эконометрическом моделировании, не позволяют использовать такие условия на

¹ Исследование выполнено при финансовой поддержке РФФИ (проект № 13-01-00749 "Качество моделей математической обработки наблюдений в социальных и экономических системах").

стадии идентификации модели, что может стать причиной многократных корректировок модели и, как следствие, затягивания процесса получения результата.

В связи с этим представляется целесообразной разработка специфических методов моделирования, нивелирующих указанные сложности. По мнению авторов, с позиций повышения эффективности проводимых исследований, наиболее перспективными являются следующие два направления: снижение "нагрузки" на этап верификации модели и достижение высокой точности аппроксимации как основного качественного показателя получаемой модели.

Сказанное означает осуществление, во-первых, уже на стадии определения параметров модели формализации в качестве ограничений тех требований, соблюдение которых предполагается контролировать на этапе верификации и, во-вторых, выбора в качестве критерия оптимальности функционала, характеризующего соответствие расчетных и экспериментальных данных.

Основу большинства математико-статистических подходов к решению эконометрических задач составляет метод наименьших квадратов [1, 2], суть которого заключается в определении параметров зависимостей, минимизирующих сумму квадратов невязок — отклонений расчетных значений эндогенной величины от экспериментальных. При этом предполагается, что невязки являются случайными величинами, подчиняются нормальному закону распределения с нулевым математическим ожиданием, гетероскедастичны и не автокоррелированы [2]. Проверка этих предположений представляет собой самостоятельную задачу, решение которой в случае ограниченного количества экспериментальной информации будет затруднено.

Критерий, используемый в методе наименьших квадратов, характеризует совокупную погрешность аппроксимации имеющихся экспериментальных данных. При этом точность аппроксимации оценивается усредненно, несмотря на то что значения некоторых невязок могут быть очень большими.

Альтернативой такому подходу к пониманию точности аппроксимации и соответствующему формированию критерия оптимальности параметров искомой зависимости является анализ значений невязок в каждом отдельном наблюдении. Примером такого подхода является метод выравнивания по Чебышеву, основная идея которого заключается в приближении экспериментальных данных таким способом, чтобы обеспечивалась равномерная точность описания во всей исследуемой области, т. е. каждая невязка по модулю не должна превышать некоторого оптимального уровня.

В настоящей работе исследована возможность применения задачи чебышевского выравнивания для идентификации моделей и показана связь оптимальных решений подобных задач с оценками для значений таких показателей совокупной погрешности аппроксимации, как сумма модулей невязок и средняя ошибка аппроксимации.

1. Чебышевское приближение

Задача чебышевского приближения системы m несовместных линейных уравнений с n неизвестными

$$\eta_i(\mathbf{x}) \equiv a_{i1}x_1 + \dots + a_{in}x_n + a_i = 0, \quad i = \overline{1, m} \quad (1)$$

состоит в том [3], чтобы определить чебышевскую точку системы (1) $\mathbf{x}^* = \{x_1^*, \dots, x_n^*\}$ такую, что

$$\max_{i = \overline{1, m}} |\eta_i(\mathbf{x}^*)| = \inf_{\mathbf{x}} \max_{i = \overline{1, m}} |\eta_i(\mathbf{x})| = \delta^*. \quad (2)$$

Соотношения (1) определяют в n -мерном пространстве систему из m плоскостей. В соответствии с (2) точка \mathbf{x}^* является наименее отклоняющейся по модулю от этой системы. Любая другая точка будет отклоняться по модулю, хотя бы от одной из плоскостей, на значение, большее δ^* .

Задача чебышевского приближения системы (1) сводится к задаче линейного программирования посредством введения дополнительной переменной λ , гарантирующей выполнение системы линейных неравенств

$$|\eta_i(\mathbf{x})| \leq \lambda, \quad i = \overline{1, m}.$$

Это позволяет, используя очевидные свойства модулей, записать следующую задачу линейного программирования:

$$\min_{\mathbf{x}, \lambda} \lambda, \quad (3)$$

$$a_{i1}x_1 + \dots + a_{in}x_n + a_i - \lambda \leq 0, \quad i = \overline{1, m}, \quad (4)$$

$$a_{i1}x_1 + \dots + a_{in}x_n + a_i + \lambda \geq 0, \quad i = \overline{1, m}. \quad (5)$$

В том случае, когда имеются ограничивающие условия на значения переменных задачи (1)–(2), заданных в виде линейных ограничений, определяющих некоторый выпуклый многогранник X в n -мерном пространстве, задача определения параметров, подобная задаче чебышевского приближения, примет вид

$$\max_{i = \overline{1, m}} |\eta_i(\mathbf{x})| = \min_{\mathbf{x} \in X} \max_{i = \overline{1, m}} |\eta_i(\mathbf{x})| = \delta^*,$$

а эквивалентная ей задача линейного программирования запишется следующим образом:

$$\min_{\mathbf{x}, \lambda} \lambda, \quad (6)$$

$$a_{i1}x_1 + \dots + a_{in}x_n + a_i - \lambda \leq 0, \quad i = \overline{1, m}, \quad (7)$$

$$a_{i1}x_1 + \dots + a_{in}x_n + a_i + \lambda \geq 0, \quad i = \overline{1, m}, \quad (8)$$

$$\mathbf{x} \in X. \quad (9)$$

Задача (3)—(5) в силу теоремы Вейерштрасса о достижении непрерывной на компакте функции своей точной нижней границы всегда имеет решение. По этой же причине, в случае если условия (9) не нарушают совместности множества допустимых решений (7)—(8), задача (6)—(9) также всегда имеет решение. Приведенным задачам можно дать следующие геометрические интерпретации. При некотором λ и при каждом фиксированном i ограничения (4) и (5) определяют в соответствующем пространстве полосу — часть пространства, ограниченную двумя плоскостями. Пересечение конечного числа таких полос может либо образовывать многогранник (область S на рис. 1, а), либо быть пустым (рис. 1, б). Поэтому геометрически задача (4)—(6) сводится к тому, чтобы найти такое λ^* , при котором множество S будет не пустым и имеющим минимальный объем (минимальную площадь в случае двух переменных).

Дополнительные условия на значения переменных, заданные в виде ограничений (9), и не противоречащие (7)—(8), "урезают" множество S или оставляют его неизменным, что, в свою очередь, может приводить к увеличению оптимального значения λ^* .

Задача чебышевского приближения системы линейных уравнений является частным случаем задачи минимизации суммы модулей линейных функций:

$$z(\mathbf{x}) = \sum_{i=1}^m |\eta_i(\mathbf{x})|. \quad (10)$$

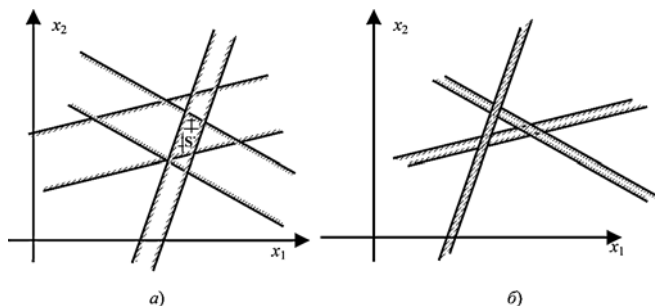


Рис. 1. Геометрическая интерпретация ограничений (4)—(5): а — непустое пересечение; б — пустое пересечение

Введем дополнительные переменные $\{\lambda_i, \lambda_i \geq 0, i = \overline{1, m}\}$ такие, чтобы выполнялись соотношения

$$|\eta_i(\mathbf{x})| \leq \lambda_i, \quad i = \overline{1, m}. \quad (11)$$

Тогда задаче (10)—(11) может соответствовать эквивалентная задача линейного программирования [3]:

$$\min_{\mathbf{x}, \{\lambda_i\}} \sum_{i=1}^m \lambda_i, \quad (12)$$

$$\lambda_i + a_{i1}x_1 + \dots + a_{in}x_n + a_i \geq 0, \quad i = \overline{1, m}, \quad (13)$$

$$\lambda_i - a_{i1}x_1 - \dots - a_{in}x_n - a_i \geq 0, \quad i = \overline{1, m}. \quad (14)$$

Из теоремы Вейерштрасса следует, что задача (12)—(14) всегда имеет решение, в том числе и при существовании дополнительных ограничений на переменные \mathbf{x} в виде линейных неравенств и (или) равенств, не противоречащих условиям (13), (14).

Если в задачу (12)—(14) ввести условие равенства всех переменных $\{\lambda_i, i = \overline{1, m}\}$, то задача линейного программирования, эквивалентная задаче определения минимума суммы модулей линейных функций, будет иметь следующий вид:

$$\min_{\mathbf{x}, \lambda} m\lambda,$$

$$\lambda + a_{i1}x_1 + \dots + a_{in}x_n + a_i \geq 0, \quad i = \overline{1, m},$$

$$\lambda - a_{i1}x_1 - \dots - a_{in}x_n - a_i \geq 0, \quad i = \overline{1, m},$$

и будет с точностью до константы m в целевой функции совпадать с задачей (3)—(5). Из этого следует очевидный вывод: оптимальное решение задачи чебышевского приближения системы линейных уравнений является допустимым решением задачи минимизации суммы модулей линейных функций, при этом оптимальное значение f^* целевой функции (12) не превышает оптимального значения λ^* целевой функции (3), увеличенного в m раз, т. е. справедливо соотношение $f^* \leq m\lambda^*$.

2. Связь идентификации эконометрической модели и задачи чебышевского приближения

Идентификация линейной модели как одного из этапов эконометрического моделирования заключается в том, чтобы на основании имеющейся экспериментальной информации $\{(\mathbf{x}_t, y_t), t = \overline{1, m}\}$ (где $\mathbf{x}_t = \{x_{1t}, \dots, x_{nt}\}$ — вектор значений экзогенных факторов в момент времени t ; y_t — значение эндогенного фактора в момент времени t ; m — число наблюдений) определить параметры $\mathbf{a} = \{a_0, a_1, \dots, a_n\}$ линейной зависимости

$$\hat{y} = a_0 + a_1x_1 + \dots + a_nx_n. \quad (15)$$

Очевидно, что $\forall t = \overline{1, m}$ — расчетные значения эндогенной переменной в моменты времени t (\hat{y}_t) должны быть близки к фактическим значениям y_t . Выражения для невязок имеют следующий вид:

$$\varepsilon_t = \hat{y}_t - y_t = (a_0 + a_1 x_1 + \dots + a_n x_n)|_t - y_t, \quad i = \overline{1, m}. \quad (16)$$

Абсолютное соответствие расчетных и экспериментальных данных эндогенной переменной означает равенство всех невязок нулю:

$$\varepsilon_t = (a_0 + a_1 x_1 + \dots + a_n x_n)|_t - y_t = 0, \quad i = \overline{1, m}.$$

Последнее соотношение с точностью до обозначений совпадает с задачей чебышевского приближения (1). Таким образом, определение параметров линейной зависимости (15) может проводиться на основе решения задачи линейного программирования вида (3)—(5), а в том случае, если исследователь считает необходимым учесть некоторые дополнительные условия, на основе решения задачи линейного программирования вида (6)—(9). Важным преимуществом такого подхода, помимо априорного учета дополнительных условий, является отсутствие требования на число наблюдений, что следует из существования решений задач линейного программирования (3)—(5) и (6)—(9) вне зависимости от параметра m .

Одним из общеупотребительных числовых критериев, позволяющих делать вывод о качестве эконометрической модели, является средняя ошибка аппроксимации [1, 2]:

$$\bar{A} = \frac{1}{m} \sum_{t=1}^m \left| \frac{\hat{y}_t - y_t}{y_t} \right|. \quad (17)$$

Величина (17) является безразмерной и допускает удобную интерпретацию в процентах. Считается, что эконометрическая модель имеет хорошую точность аппроксимации, если \bar{A} не превышает 8...10 % [2].

Справедливо следующее.

Утверждение. В случае идентификации модели линейной структуры (15), задача определения ее параметров, минимизирующих выражение (17), эквивалентна задаче минимизации суммы модулей линейных функций (1), (10).

Доказательство. Задача определения точного вида модели (15) в соответствии с критерием (17) заключается в определении на основании экспериментальной информации $\{(x_p, y_t), t = \overline{1, m}\}$ значений параметров $\{a_0, a_1, \dots, a_n\}$, минимизирующих функционал (17); $\hat{y}_t = \hat{y}(x_1|_t, \dots, x_n|_t)$, где $x_j|_t$ — экспериментальное значение j -й экзогенной переменной в момент времени t .

Для характеристики близости расчетных и экспериментальных значений фактора y будем использовать значения невязок (16).

Умножим невязки ε_t на соответствующие константы $\frac{1}{my_t}$ и преобразуем полученные выражения следующим образом:

$$\begin{aligned} \varepsilon_t \frac{1}{my_t} &= a_0 \frac{1}{my_t} + a_1 \frac{x_1|_t}{my_t} + \dots + a_n \frac{x_n|_t}{my_t} - \frac{y_t}{my_t} = \\ &= a_0 \frac{1}{my_t} + a_1 \frac{x_1|_t}{my_t} + \dots + a_n \frac{x_n|_t}{my_t} - \frac{1}{m}. \end{aligned}$$

Введем следующие обозначения:

$$b_{t0} = \frac{1}{my_t}; \quad b_{t1} = \frac{x_1|_t}{my_t}, \dots, \quad b_{tn} = \frac{x_n|_t}{my_t}; \quad b_t = -\frac{1}{m}, \quad t = \overline{1, m}.$$

Тогда задача идентификации модели (15) может быть сформулирована следующим образом:

$$\min_{a_0, a_1, \dots, a_n} \bar{A} = \min_{a_0, a_1, \dots, a_n} \sum_{t=1}^m |\hat{\varepsilon}_t(a_0, a_1, \dots, a_n)|, \quad (18)$$

$$\begin{aligned} \hat{\varepsilon}_t(a_0, a_1, \dots, a_n) &\equiv b_{t0} a_0 + b_{t1} a_1 + \dots + b_{tn} a_n + b_t, \\ &t = \overline{1, m}. \end{aligned} \quad (19)$$

Задача (18)—(19) с точностью до обозначения переменных совпадает с задачей минимизации суммы модулей линейных функций (1), (10). Утверждение доказано.

Заметим, что значение средней ошибки аппроксимации очевидным образом ограничивается сверху:

$$\bar{A} = \frac{1}{m} \sum_{t=1}^m \left| \frac{\hat{y}_t - y_t}{y_t} \right| \leq \frac{1}{m \min_{t=1, m} \{|y_t|\}} \sum_{t=1}^m |\hat{y}_t - y_t|.$$

Таким образом, с учетом ранее доказанного неравенства $f^* \leq m\lambda^*$ для идентифицируемой модели линейного вида справедливо соотношение

$$\bar{A}^* \leq \frac{f^*}{m \min_{t=1, m} \{|y_t|\}} \leq \frac{\lambda}{\min_{t=1, m} \{|y_t|\}}, \quad (20)$$

где \bar{A}^* — оптимальное значение критерия (18).

На практике соотношение (20) может давать достаточно грубую оценку \bar{A}^* (см. ниже п. 3 и пп. 4.2). Однако если по результатам решений задач определения параметров линейных моделей с помощью сведения к задачам чебышевского приближения и минимизации суммы модулей линейных функций (в том числе и при наличии дополнительных условий) рассчитывать средние ошибки аппроксимации ($\bar{A}^{\text{чеб}}$ и $\bar{A}^{\text{мод}}$ соответственно), то будет выполняться соотношение

$$\bar{A}^* \leq \bar{A}^{\text{мод}} \leq \bar{A}^{\text{чеб}}, \quad (21)$$

Таблица 1

**Результаты идентификации
линейной модели урожайности зерновых культур**

Модель	Крите- рий	Характеристики		
		R^2	F	\bar{A} , %
5 экзогенных факторов				
Задача чебышевского приближения с дополнительными условиями: $\hat{y} = 4,130 + 19,332x_2 + 3,945x_4 - 5,010x_5$	$\lambda^* = 2,258$	0,442	6,73	14,99
Задача минимизации суммы модулей невязок: $\hat{y} = 6,665 + 0,445x_1 + 4,560x_4 - 1,330x_5$	$f^* = 17,273$	0,460	7,24	8,32
Задача минимизации средней ошибки аппроксимации: $\hat{y} = 6,504 + 0,435x_1 + 1,562x_2 + 2,943x_4 - 0,645x_5$	$\bar{A}^* = 0,081$	0,464	7,38	8,09
2 экзогенных фактора (x_3 и x_4)				
Задача чебышевского приближения с дополнительными условиями: $\hat{y} = 8,858 + 0,185x_3 + 2,353x_4$	$\lambda^* = 2,519$	0,430	6,41	17,00
Задача минимизации суммы модулей невязок: $\hat{y} = 6,754 + 0,336x_3 + 3,777x_4$	$f^* = 18,735$	0,474	7,66	9,01
Задача минимизации средней ошибки аппроксимации: $\hat{y} = 6,807 + 0,372x_3 + 2,425x_4$	$\bar{A}^* = 0,086$	0,439	6,65	8,59

справедливость которого следует из смысла \bar{A}^* и из того факта, что задача чебышевского приближения является частным случаем задачи минимизации суммы модулей линейных функций, и поэтому $\bar{A}^{\text{мод}} \leq \bar{A}^{\text{чeб}}$.

Таким образом, результаты решения задачи чебышевского приближения позволяют получить оценку лучшего значения средней ошибки аппроксимации — критерия близости расчетных и экспериментальных данных (табл. 1).

3. Моделирование линейных зависимостей

Возможности описанного подхода продемонстрируем на примере, рассмотренном в работе [1], посвященном исследованию зависимости урожайности зерновых культур (y , ц/га) от следующих пяти переменных:

- число тракторов приведенной мощности на 100 га (x_1 , ед./100 га);
- число зерноуборочных комбайнов на 100 га (x_2 , ед./100 га);
- число орудий поверхностной обработки почвы на 100 га (x_3 , ед./100 га);

- количество удобрений, расходуемых на гектар (x_4 , т/га);
- количество химических средств защиты растений, расходуемых на гектар (x_5 , ц/га).

Исходя из имеющихся 20 наблюдений по методу наименьших квадратов была получена следующая линейная зависимость:

$$\hat{y} = 3,515 - 0,006x_1 + 15,542x_2 + 0,110x_3 + 4,475x_4 - 2,932x_5. \quad (22)$$

Линейная зависимость (22) имеет недостаточно хорошие характеристики: коэффициент детерминации $R^2 = 0,517$, F -критерий Фишера $F = 3,001$, средняя ошибка аппроксимации $\bar{A} = 10,42\%$, t -статистика значима только у коэффициента при переменной x_4 . В дальнейшем, посредством исключения из рассмотрения "несущественных" экзогенных переменных была получена зависимость

$$\hat{y} = 7,29 + 3,48x_3 + 3,48x_4$$

со статистически значимыми коэффициентами и следующими характеристиками: $R^2 = 0,482$, $F = 7,921$, $\bar{A} = 10,47\%$.

Относительно приведенного примера обращает внимание тот факт, что параметры при первых трех переменных в идентифицируемой линейной модели (15), очевидно, не могут быть отрицательными ввиду того, что для обеспечения высокой урожайности важен требуемый объем работ, а не количество технических средств труда, которыми этот объем будет выполнен.

Исходя из этого при формировании системы ограничений задач линейного программирования на значения искомым параметров были наложены дополнительные условия: $a_1 \geq 0$, $a_2 \geq 0$, $a_3 \geq 0$.

Заметим, что приведенные результаты подтверждают справедливость соотношений (20)¹ и (21). Соотношение (20) для случая зависимости с пятью и с двумя переменными имеет вид $0,081 \leq 0,125 \leq 0,327$ и $0,086 \leq 0,136 \leq 0,365$ соответственно.

4. Идентификация нелинейных моделей

4.1. Формализация задач

Если спецификация модели $\hat{y} = \hat{y}(x, a)$ не является линейной, то определение параметров a может происходить на основе решения задачи математического программирования:

$$\min_{a, \lambda} \lambda, \quad (23)$$

$$|\varepsilon_i| \leq \lambda, \quad i = \overline{1, m}, \quad (24)$$

$$a \in A, \quad (25)$$

¹ В рассматриваемом примере $\min_{i = \overline{1, m}} \{y_i\} = 6,9$.

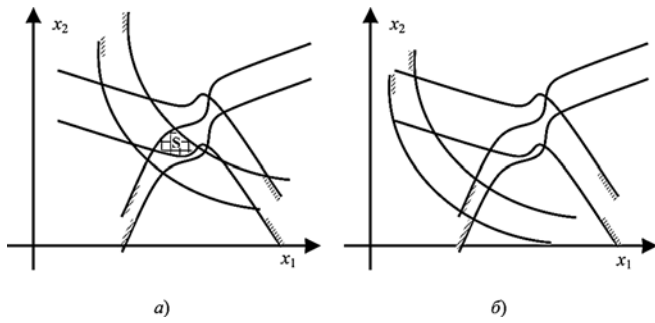


Рис. 2. Геометрическая интерпретация ограничений (24): а — непустое пересечение; б — пустое пересечение

где по аналогии с приведенными выше задачами (3)—(5) и (6)—(9) ограничения (24) характеризуют условия на значения невязок в каждом наблюдении, а ограничения (25), суть — формализованные требования, подлежащие учету.

Для случая, когда невязки (16) представляют собой монотонные функции, каждое ограничение (24) геометрически представляет собой часть пространства соответствующей размерности между поверхностями, задаваемыми уравнениями одинаковой спецификации и отличающимися лишь свободными константами (рис. 2). Поэтому если ограничения задачи (23)—(25) являются совместными, то, очевидно, что ее решение существует.

Если же в качестве критерия оптимальности будет выбрана сумма модулей невязок, то задача определения параметров идентифицируемой зависимости примет вид

$$\min_{\mathbf{a}, \{\lambda_t\}} \sum_{t=1}^m \lambda_t \quad (26)$$

$$|\varepsilon_t| \leq \lambda_t, \quad t = \overline{1, m},$$

$$\mathbf{a} \in A.$$

При этом, учитывая отсутствие привязки формулы для вычисления средней ошибки аппроксимации (17) к спецификации модели, соотношения, аналогичные (20), также будут выполняться, с той лишь разницей, что f^* — оптимальное значение критерия (26), λ^* — оптимальное значение критерия оптимизации в задаче (23)—(25), а \bar{A}^* — оптимальное значение целевой функции соответствующей задачи определения параметров идентифицируемой нелинейной функции, в которой средняя ошибка аппроксимации является критерием минимизации. В этом случае величина λ^* , строго говоря, не является решением задачи чебышевского приближения, но ее смысл остается прежним: она показывает минимальное значение всех возможных максимальных по модулю невязок.

4.2. Моделирование нелинейных зависимостей

Существенные трудности в применении математико-статистических методов для определения модельных параметров возникают как при недостаточном количестве экспериментальных данных в расчете на каждый искомый параметр, так и в ситуации, когда невозможно осуществить линеаризацию модели с нелинейной спецификацией. Описанные в настоящей работе подходы позволяют определять параметры модели и при наличии такого рода проблем. Для подтверждения сказанного рассмотрим пример моделирования гипотетического временного ряда (табл. 2, рис. 3).

По результатам визуального анализа уровней временного ряда (рис. 3) была установлена возможность использования следующей спецификации модели:

$$\hat{y} = a \cdot t^b + c. \quad (27)$$

Модель (27) не допускает осуществления линеаризации, а количество имеющихся данных (10 на три оцениваемых параметра) является недостаточным для получения значимых характеристик, рассчитываемых в рамках математико-статистических методов.

В качестве дополнительного ограничения на значения параметров модели (27) использовалось формализованное представление требования на уровень временного ряда в момент времени $t = 12$: $9,2 \leq \hat{y}|_{12} \leq 9,5$. (Такого рода условия могут формироваться на основе объективных данных или мнений экспертов, и их априорный учет может повысить степень доверия к результирующим моделям.)

Результаты проведенных расчетов (табл. 3, рис. 3) свидетельствуют о практически одинаковой значимости каждой из идентифицированных моделей, что предоставляет исследователю определенную свободу выбора, и может быть особенно существенным в случае получения дополнительной инфор-

Таблица 2

Исходные данные для моделирования временного ряда

t	1	2	3	4	5	6	7	8	9	10
y_t	1,6	3,6	3,7	5,4	6,1	6,2	6,6	8,0	8,3	8,7

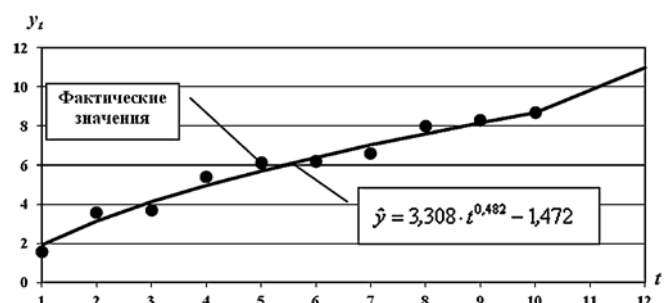


Рис. 3. Геометрическая интерпретация модели временного ряда

Результаты идентификации модели временного ряда $\hat{y} = a \cdot t^b + c$

Модель	Критерий	Характеристики			
		R^2	F	\bar{A} , %	$\hat{y} _{12}$
Задача чебышевского выравнивания с дополнительными условиями: $\hat{y} = 3,308 \cdot t^{0,482} - 1,472$	$\lambda^* = 0,449$	0,975	79,51	7,18	9,5
Задача минимизации суммы модулей невязок: $\hat{y} = 7,302 \cdot t^{0,295} - 5,704$	$f^* = 2,617$	0,974	75,35	5,25	9,5
Задача минимизации средней ошибки аппроксимации: $\hat{y} = 8,299 \cdot t^{0,269} - 6,699$	$\bar{A}^* = 0,052$	0,973	72,83	5,17	9,5

мации о значениях параметров модели или ее переменных.

Оценки (20) для полученных моделей выполняются: $0,052 \leq 0,164 \leq 0,281$.

5. Направление развития исследований

Результаты проведенных расчетов по идентификации моделей (п. 3, пп. 4.2) свидетельствуют о существовании множества наборов значений параметров, обеспечивающих приемлемые качественные характеристики моделей. В связи с этим является целесообразным проведение на основе использования методов математического моделирования дополнительных исследований, направленных на определение лучшего из числа таких наборов. При этом очевидно, что в зависимости от предъявляемых к модели требований, вполне возможна ситуация, при которой оптимальными будут являться не единичные значения параметров, а некоторые интервалы их значений. В этом смысле совокупность оптимальных значений параметров идентифицируемой модели может трактоваться как "область неопределенности" параметров модели, а окончательный вид модели может быть выявлен на основе индивидуальных предпочтений исследователя и (или) с учетом каких-либо дополнительных требований при непосредственном использовании необходимых численных методов и программных продуктов.

В приведенном контексте понимание "областей неопределенности" параметров эконометрических моделей тесно переплетается с идеями, высказанными в работе [4]. Применение подхода Л. В. Канторовича в сочетании со специально разработанными моделями (аналогичными приведенным в настоящей работе), алгоритмами и программными продуктами позволило авторам [5–8] решить проблему идентификации модели системной динамики численности населения Российской Федерации.

Большое разнообразие практических задач далеко не всегда позволяет использовать математико-статистические методы при построении эконометрических моделей, что обусловлено в том числе следующими причинами:

- применение данных методов детально разработано для линейных и линеаризуемых моделей;
- ограниченное число наблюдений не позволяет получать значимые характеристики для идентифицируемой модели;
- наличие большого количества требований, на соответствие которым модель должна проверяться на этапе ее верификации, части из которых она может не удовлетворять.

При этом необходимо принимать во внимание и объективно существующие трудности, связанные с возможной неточностью исходных данных, что можно объяснить например, многостадийностью процесса сбора информации, существенной долей субъективизма, а в некоторых случаях и умышленным искажением информации с целью "приукрасить" реальное положение дел. Оценить точность полученных наблюдений чаще всего не представляется возможным в силу отсутствия эталонных значений и возможности многократного проведения наблюдений в одних и тех же условиях, а это, в свою очередь, может стать причиной, по которой будет построена "точная модель неточных данных".

Сказанное обуславливает необходимость разработки специальных методов идентификации эконометрических моделей. И одним из возможных подходов к реализации этого является использование идей задач чебышевского приближения. Приведенные в работе постановки задач позволяют определять параметры эконометрических зависимостей с учетом требований, предъявляемых как к самим моделям, так и к составляющим их элементам. При этом существует возможность оценки степени соответствия экспериментальных и расчетных данных, определяемой показателем средней ошибки аппроксимации. Следует особо отметить, что приведенные в работе способы определения параметров идентифицируемых моделей позволяют учитывать условия на будущие значения эндогенных переменных. Это является существенным при анализе факторов, имеющих высокую инерционность, в силу чего для них может быть оценен диапазон возможных значений в краткосрочной перспективе.

Приведенные в работе расчеты в силу достаточно хороших качественных характеристик позволяют не только констатировать целесообразность использования описанного подхода для практического применения, но и могут составить основу для проведения более детальных исследований, направленных на определение точного вида модели.

Список литературы

1. Айвазян С. А., Мхитарян В. С. Прикладная статистика и основы эконометрики. М.: ЮНИТИ, 1998. 1022 с.
2. Эконометрика: учебник / Под ред. И. И. Елисеевой. М.: Финансы и статистика, 2008. 575 с.
3. Зуховицкий С. И., Авдеева Л. И. Линейное и выпуклое программирование. М.: Наука, 1967. 460 с.
4. Канторович Л. В. О некоторых новых подходах к вычислительным методам и обработке наблюдений // Сибирский математический журнал. 1962. Т. 3, № 5. С. 701–709.

5. Спивак С. И., Кантор О. Г. Построение модели системной динамики в условиях ограниченной экспертной информации // Информатика и ее применения. 2014. Т. 8, № 2. С. 111–121.
6. Спивак С. И., Кантор О. Г. Риск ошибочных прогнозных оценок параметров социально-экономических систем // Управление риском. 2013. № 1. С. 2–6.
7. Спивак С. И., Кантор О. Г., Юсупова Г. Н. Алгоритм распараллеливания в задачах перебора значений многомерных величин на основе декомпозиции по данным // Системы управления и информационные технологии. 2014. № 2 (56). С. 52–57.
8. Kantor O. G., Spivak S. I. Interval Estimation of System Dynamics Model Parameters // International Journal of Applied Engineering Research. 2014. V. 9, no. 19. P. 5689–5695.

S. I. Spivak, Head of Department of Mathematical Modeling; semen.spivak@mail.ru,
Bashkir State University, Ufa,

O. G. Kantor, Senior scientist; o_kantor@mail.ru,

Institute of Social and Economic Research Ufa Research Centre Russian Academy of Sciences, Ufa

Chebyshev Approximation in Problems of Econometric Modeling

A wide variety of practical problems is not always permit to use mathematical and statistical methods in the construction of econometric models. This may be due, for example, by the following reasons: the use of these methods in detail designed for linear and linearizable models; a limited number of observations do not allow to get meaningful statistical characteristics. In addition, the resulting model should meet the requirements ensuring its adequacy. In the event of any breach of the adequacy researcher have to change the form of the function and carry out a complete cycle of remodeling.

The possibility of using the Chebyshev approximation for identification of econometric models is studied at the paper. The connection between the optimal solution of Chebyshev approximation with an average error of approximation — the criterion of model accuracy is shown. The proposed approaches to the estimation of econometric models parameters allow for researcher to take into account a number of objectively existing problems and consider the important requirements at the stage of identification directly.

Keywords: model identification, model verification, Chebyshev approximation, average approximation error, uncertainty sets of the model parameters

References

1. Ajvazjan S. A., Mhitarjan V. S. *Prikladnaja statistika i osnovy jekonometriki*. M.: JuNITI, 1998. 1022 p.
2. *Jekonometrika*: Uchebnik / Pod red. I. I. Eliseevoj. M.: Finansy i statistika, 2008. 575 p.
3. Zuhovickij S. I., Avdeeva L. I. *Linejnoe i vypukloe programirovanie*. M.: Nauka, 1967. 460 p.
4. Kantorovich L. V. O nekotoryh novyh podhodah k vychislitel'nym metodam i obrabotke nabljudenij. *Sibirskij matematicheskij zhurnal*. 1962, vol. 3, no. 5, pp. 701–709.

5. Spivak S. I., Kantor O. G. Postroenie modeli sistemnoj dinamiki v uslovijah ogranichennoj jekspertnoj informacii, *Informatika i ee primenenija*, 2014, vol. 8, no. 2, pp. 111–121.
6. Spivak S. I., Kantor O. G. Risk oshibochnyh prognoznyh ocenok parametrov social'no-jekonomicheskikh sistem, *Upravlenie riskom*, 2013, no. 1, pp. 2–6.
7. Spivak S. I., Kantor O. G., Jusupova G. N. Algotnm raspallelivanija v zadachah perebora znachenij mnogomernyh velichin na osnove dekompozicii po dannym, *Sistemy upravlenija i informacionnye tehnologii*, 2014, no. 2 (56), pp. 52–57.
8. Kantor O. G., Spivak S. I. Interval Estimation of System Dynamics Model Parameters, *International Journal of Applied Engineering Research*, 2014, vol. 9, no. 19, pp. 5689–5695.

УДК 621.372.061

В. В. Ракитин^{1, 2}, канд. техн. наук, ст. науч. сотр.,
С. Г. Русаков¹, чл.-корр. РАН, проф., гл. науч. сотр., gusakov@ippm.ru

¹ Институт проблем проектирования в микроэлектронике РАН, Москва

² ФГУП "НИИ Физических проблем им. Ф. В. Лукина", Москва

Генераторные схемы на мемристорных элементах

Обсуждается состояние разработок генераторных схем на мемристорных приборах. Показаны особенности конструирования автогенераторных схем при использовании свойств мемристоров. Рассмотрены базовые схемотехнические решения безиндуктивных и безреактивных генераторов, а также хаотических генераторов на мемристорных приборах.

Ключевые слова: мемристоры, мемристорные системы, мемристорные генераторы, безиндуктивные генераторы, безреактивные генераторы, хаотические генераторы

Введение

Ожидание коммерческого внедрения интегральных технологий с мемристорными приборами ускорило разработку схем на мемристорах в разных областях применения. В первую очередь, эти разработки направлены на достижения новых качественных показателей в схемах памяти. Благодаря свойствам мемристоров новые схемотехнические принципы получают в ближайшее время распространение и в других сферах применения цифровой и аналоговой схемотехники.

Новое направление нелинейной теории цепей составляют генераторные и автогенераторные схемы на мемристорных приборах.

В настоящей статье обсуждается начальная стадия разработки новых схемотехнических решений для мемристорных генераторов с использованием основных свойств мемристоров.

Мемристор — пассивный двухполюсник с сопротивлением, изменяющимся в зависимости от протекающего через него заряда. Фактически это резистор с аналоговой памятью. Термин "мемристор" предложен в 1971 г. проф. Л. Чуа для описания прибора, электрический заряд которого связан с магнитным потоком [1]. В последующем термин расширен на более широкий класс мемристорных систем [2]. В 2008 г. в лаборатории фирмы Хьюлетт-Паккард был создан мемристорный прибор со структурой металл—диэлектрик—металл [3]. С этого момента исследования мемристорных элементов и систем на их основе стало одним из многообещающих направлений электроники [4—6]. Мемристоры еще находятся на стадии экспериментальных исследований, но широкий фронт работ и успехи в техно-

логии их изготовления позволяют надеяться на появление новых изделий на их основе.

Простота конструкции мемристоров и совместимость технологии их изготовления с кремниевой КМОП-технологией являются важными обстоятельствами, способствующими будущему внедрению мемристоров. Причем сами мемристоры могут формироваться на завершающей стадии изготовления ИС в виде специальных контактов между слоями металлов. Области применения мемристоров могут быть как различные системы хранения информации, так и специальные схемы, обладающие возможностями удобной регулировки и самонастройки.

Генераторы на мемристорах являются новым классом нелинейных схем, которые могут быть использованы в различных приложениях [7]. Используя инерционные свойства мемристоров, можно построить различного типа генераторы сигналов.

Ниже рассмотрены основные принципы построения мемристорных генераторов с учетом свойства аналоговой памяти мемристоров.

В разд. 1 и 2 содержатся предварительные сведения о типах генераторов на мемристорах и модели мемристорного прибора. Особенности построения генераторов на мемристорах, включая безиндуктивные мемристорные генераторы и безреактивные генераторы, обсуждаются в разд. 3. В разд. 4 рассмотрены хаотические генераторы на мемристорах.

1. Предпосылки

К настоящему времени уже предложены различные варианты генераторов на мемристорах. Представлены так называемые безреактивные мемристорные генераторы, в которых мемристоры заме-

няют конденсаторы [8, 9]. Предложены генераторы, в которых мемристоры заменяют резисторы, расширяя функциональные возможности исходных генераторов. Однако в них присутствуют реактивные элементы, требования к которым могут быть несовместимы с интегральной реализацией устройств. Предложены различные типы таких генераторов: с мостом Вина [10], фазосдвигающие [11], моностабильные [12]. Большое внимание уделяется генераторам хаотических сигналов [13–15]. На основе мемристорных генераторов предполагается создание искусственных нейронов нового поколения [16], в том числе разработка моделей нейронов с синаптической пластичностью [17], а в перспективе построение мемристорных осцилляторных нейрокомпьютеров [18].

Условно мемристорные генераторы можно разделить на два класса (рис. 1). К первому классу следует отнести автогенераторы, свойства которых в основном определяются самими мемристорами, дополненными вспомогательными схемами на основе стандартных элементов, задающими режим работы мемристора. Они не содержат реактивных элементов и удобны в интегральном исполнении (рис. 1, а).

Второй класс мемристорных генераторов [10] можно разделить на ряд типов схем, в том числе:

- автономные, т. е. с питанием от источника постоянного тока;
- неавтономные с внешней накачкой, подсоединенные к источнику внешнего сигнала.

На рис. 1, б внешний сигнал $E(t)$ зависит от времени.

В составе этих генераторов наряду с линейными реактивными элементами могут присутствовать и другие нелинейные приборы. В общем случае может быть нелинейным сам источник питания $E(i)$.

Второй класс генераторов является универсальным по возможному целевому назначению: гармонические генераторы, генераторы специальной формы, хаотические генераторы. Принято характеризовать генераторы этого класса числом степеней свободы — суммарным числом конденсаторов, индуктивностей и самих мемристоров. Важным свойством этого класса является возможность хаотиче-

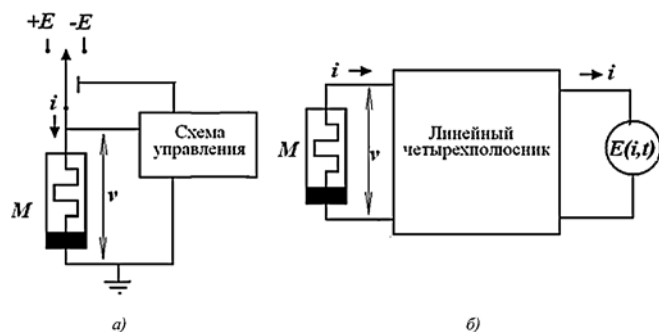


Рис. 1. Структурные схемы релаксационного (а) и универсального (б) генераторов на мемристорах

ских колебаний при числе степеней свободы 3 и выше при наличии нелинейного элемента и элемента с отрицательным дифференциальным сопротивлением (ОДС).

Анализируя состояние разработок мемристоров и генераторных систем на их основе, необходимо учитывать, что с начальной стадией их развития связаны следующие условия и ограничения:

- отсутствие общепринятой системы параметров и методик их измерений;
- сложность процессов, протекающих в мемристорах, и, как следствие, сложность их математического описания;
- отсутствие данных о разбросе и стабильности параметров мемристоров;
- многообразие типов поведения мемристоров;
- разнообразие математических моделей мемристоров.

2. Модель мемристора

Поведение мемристора описывается двумя характеристиками: связью между током i и напряжением v — "мгновенной" вольт-амперной характеристикой (ВАХ), зависящей от его состояния w , и изменением состояния под действием входного управляющего сигнала (тока или напряжения) — управляющей характеристикой (УХ) [19]. В первом случае имеем мемристор, управляемый током, во втором случае — мемристор, управляемый напряжением. В общем случае мемристор, управляемый током, описывается следующими соотношениями:

$$v = R(w, i)i, \quad (1)$$

$$\frac{dw}{dt} = F(w, i). \quad (2)$$

Мемристор с управлением напряжением описывается системой

$$i = G(w, v)v, \quad (3)$$

$$\frac{dw}{dt} = F(w, v). \quad (4)$$

Функция $R(w, i)$ имеет размерность сопротивления и называется мемристансом. Функция $G(w, v)$ имеет размерность проводимости и называется мемкондактансом. Функция управления состоянием $F(w, v)$ при безразмерной переменной w имеет размерность, обратную временной переменной, и определяет скорость изменения состояния.

В первоначальном варианте [1] мемристор, управляемый зарядом (в нашем случае состоянием), описывался более простыми соотношениями, чем (1)–(2):

$$v = R(w)i, \quad (5)$$

$$\frac{dw}{dt} = i. \quad (6)$$

Чтобы проводить анализ более широкого круга приборов, в работе [2] введено понятие мемристорной системы. В этом случае поведение описывается формулами (5) и (2):

$$v = R(w)i,$$

$$\frac{dw}{dt} = F(w, i).$$

Конкретный вид приведенных выше функций определяется конструкцией физических приборов. Так, в линейно-дрейфовой модели мемристоров [3] обе функции в формулах (1), (2) линейны по каждой из переменных. В результате система (1), (2) или (5), (6) принимает вид

$$v = (R_{on}w + R_{off}(1 - w))i; \quad (7)$$

$$\frac{dw}{dt} = kwi, \quad (8)$$

где R_{on} — минимальное сопротивление мемристора; R_{off} — максимальное сопротивление мемристора; k — коэффициент инерционности мемристора.

Поскольку сама переменная состояния w ограничена, то в формулу УХ (6) или (8) вводят множителем специальную оконную функцию $F_O(w)$, которая равна нулю вне интервала $0 \leq w \leq 1$. Внутри интервала эта функция, в частности, может равняться 1.

При анализе электрических схем с мемристорами можно не вводить переменную состояния, а задать инерционность изменения сопротивления, и система (5), (6) примет вид

$$v = Ri; \quad (9)$$

$$\frac{dR}{dt} = k(R_{on} - R_{off})i = k_R i. \quad (10)$$

В некоторых теоретических работах на функции $R(w, i)$ — мемристанса и $G(w, v)$ — мемкондуктанса накладывают требования непрерывности, монотонности и взаимно однозначности.

3. Особенности построения генераторных схем на мемристорах

3.1. Безреактивные релаксационные генераторы

Генераторы электрических сигналов различной формы широко применяют в самых разных приложениях. Наличие мемристоров открывает новые возможности реализации простых низкочастотных генераторов и генераторов специального типа (с автоподстройкой, хаотических колебаний и др.). Особое значение имеют генераторы без конденсаторов и индуктивностей, формирование которых в составе ИС затруднено или требует большой площади на кристалле. Такие схемы получили название безреактивных генераторов. Они могут использоваться

для формирования как релаксационных, так и синусоидальных колебаний.

В составе интегральных схем наиболее просто реализуются генераторы на основе RC -схем и кольцевые генераторы. RC -генератор может содержать конденсаторы, выполняющие функции интеграторов. Поскольку мемристор также интегрирует входной сигнал, то возможно рассмотрение вопроса о замене конденсаторов на мемристоры. Однако ситуация с мемристорами является более сложной, поскольку в мемристоре интеграл входного сигнала определяет не сам выходной сигнал, а внутренний параметр (мемристанс или мемкондуктанс). Поэтому проектирование генераторов на мемристорах требует специального рассмотрения.

Рассмотрим возможность замены мемристором интегрирующего конденсатора в простейшем релаксационном генераторе.

Известно большое число вариантов релаксационных генераторов с использованием конденсаторов в качестве интегрирующих элементов. Основной принцип этих генераторов — заряд конденсатора током до определенного потенциала V_{cm} с последующим разрядом конденсатора током противоположной полярности до потенциала V_{cm} , после чего полярность тока снова изменяется. Минимально простой генератор такого типа — триггер Шмитта с конденсатором в цепи отрицательной обратной связи (рис. 2, а). Передаточная характеристика триггера Шмитта (рис. 2, б) при высоком входном напряжении скачком меняет знак выходного напряжения, а при большом отрицательном напряжении скачком делает выходное напряжение положительным. Это обеспечивает непрерывное изменение напряжения на конденсаторе и прямоугольные импульсы на выходе триггера Шмитта.

Мемристор также является интегрирующим элементом. При перемене знака заряжающего тока его состояние обратимо изменяется. Однако в отличие от конденсатора у мемристора изменение знака входного сигнала приводит не только к изменению состояния (значению его сопротивления), но и к изменению полярности выходного сигнала. Поэтому для построения генератора с мемристором в качестве интегратора требуется специальный компаратор с более сложной передаточной функцией — двухпороговый компаратор.

Возможны различные варианты релаксационных мемристорных генераторов, в том числе содержащие один и более мемристоров. Два таких генератора представлены на рис. 2, в и рис. 2, д.

Генераторы содержат интегрирующую цепь, подключенную между выходом компаратора и общим (нулевым) потенциалом. Она содержит мемристор M и резистор R (возможно и два мемристора). На нее подается выходное напряжение компаратора V_o и напряжение с ее средней точки V_i поступает на вход компаратора. Если к выходу компаратора

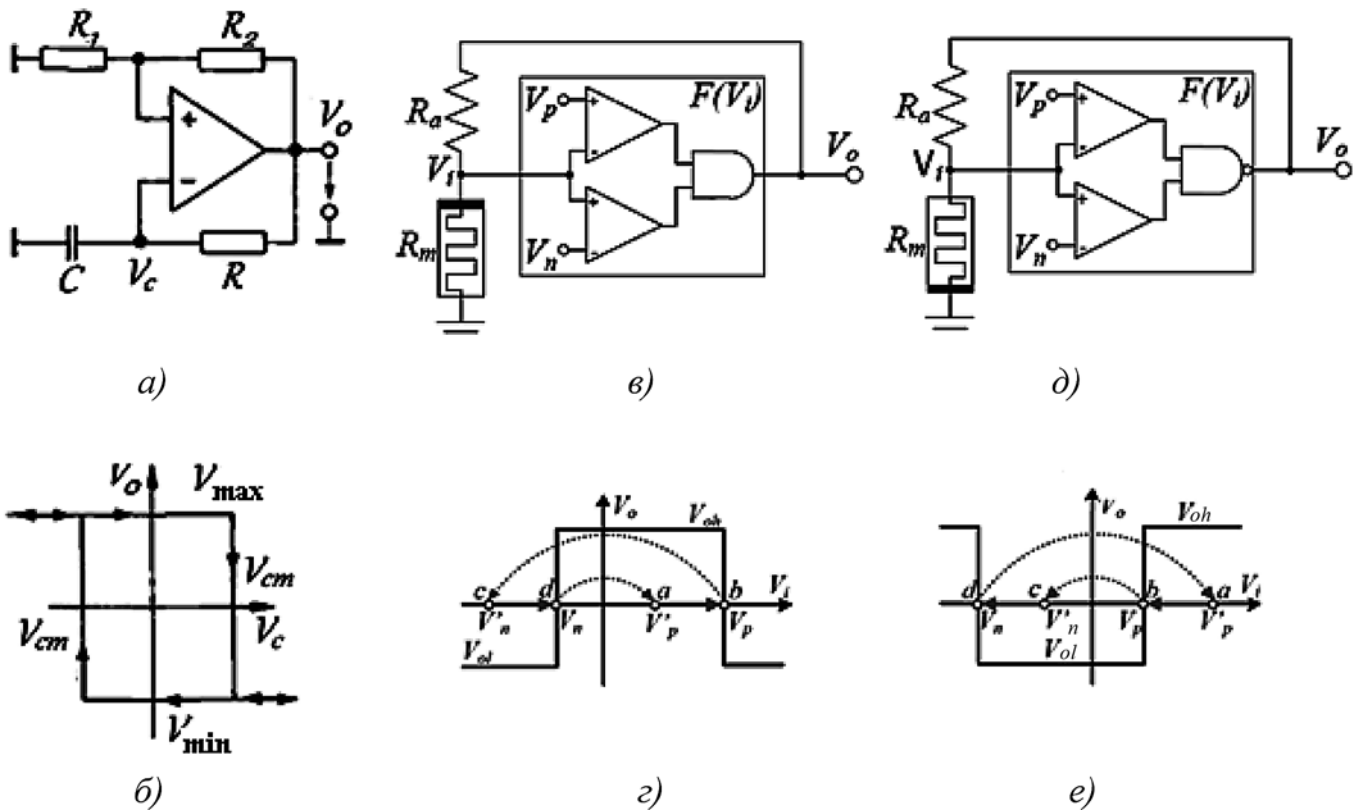


Рис. 2. Релаксационные генераторы:

a — на триггере Шмитта; *б* — передаточная характеристика триггера Шмитта; *в* — положительный $R-M$ -генератор; *г* — положительная передаточная характеристика; *д* — отрицательный $R-M$ -генератор; *е* — отрицательная передаточная характеристика [9]

подключен резистор, а мемристор подключен к общей точке положительным электродом — имеем $R-M$ положительный интегратор (рис. 2, *в*). У него при положительном напряжении V_o сопротивление мемристора и потенциал V_i возрастают. Если мемристор подключен к общей точке отрицательным электродом, имеем $R-M$ — отрицательный интегратор (рис. 2, *д*). У него при положительном напряжении V_o сопротивление мемристора и потенциал V_i падают. Возможны и другие варианты интеграторов: $M-R$ и $M-M$.

Двухпороговый компаратор может быть положительным или отрицательным, в зависимости от типа функции передаточной характеристики (рис. 2, *г* и *е*). Положительный двухпороговый компаратор содержит два простых компаратора, соединенных по схеме И (рис. 2, *в*), а отрицательный — по схеме И-НЕ (рис. 2, *д*). В результате при входном сигнале V_i , лежащем в интервале между нижним V_n и верхним порогом V_p , на выходе положительного компаратора будет высокий уровень V_{oh} , а при сигнале вне интервала — низкий уровень выходного сигнала V_{ol} (рис. 2, *г*). У отрицательного компаратора ситуация противоположная (рис. 2, *е*). На рис. 2 типы интеграторов и компараторов совпадают, и самим генераторам присвоен тот же тип.

Подобная конфигурация генераторов обеспечивает периодическое изменение напряжения V_i , по-

казанное стрелками на рис. 2, *г* и рис. 2, *е*. Покажем это на примере положительного $R-M$ -генератора. Пусть первоначально сопротивление мемристора R таково, что $V_n < V_i < V_p$, тогда поскольку $V_o = V_{oh} > 0$ сопротивление R , а значит и V_i возрастает. При достижении верхнего порога ($V_i = V_p$) компаратор изменяет знак выходного напряжения V_o , и напряжение V_i тоже становится отрицательным (V'_n на рис. 2, *г*). При отрицательном V_o сопротивление R падает, а отрицательное V_i возрастает. При достижении нижнего порога ($V_i = V_n$) компаратор снова изменяет знак выходного напряжения, и сигнал V_i оказывается в диапазоне $V_n < V_i < V_p$, т. е. ситуация возвращается к исходному состоянию. Процесс далее будет периодически повторяться.

Количественное поведение конкретного мемристорного генератора описывается системой уравнений Кирхгофа, содержащих модель мемристора (9), (10) и передаточную функцию компаратора. Для варианта, изображенного на рис. 2, *в*, из равенства токов в положительной интегрирующей цепочке вытекает следующее соотношение:

$$\frac{V_i}{R_m} = \frac{V_o(V_i)}{R_m + R_a}, \quad (11)$$

где $V_o(V_i)$ — передаточная функция положительного компаратора, принимающая значения V_{oh} и V_{ol} .

Условия и параметры генерации [9]

Параметры	Тип генератора		
	<i>M—M</i>	<i>M—R</i>	<i>R—M</i>
Зависимость от начальных условий	Да	Нет	Нет
Частота генерации	$\frac{k' V_{oh}^2}{2(V_p + V_n)(R_{ai} + R_{bi})^2}$	$\frac{k' V_n^2 V_p^2}{R_a^2 V_{oh}(V_p^2 - V_n^2)}$	$\frac{k'(V_{oh} - V_p)^2(V_{oh} + V_n)^2}{R_a^2 V_{oh}(V_p + V_n)(2V_{oh} - V_p + V_n)}$
Минимальная частота	$\frac{k' \max(V_p^2, (V_{oh} + V_n)^2)}{2R_{off}^2(V_p + V_n)}$	$\frac{k' V_p^2(V_{oh} + V_n)^2}{R_{off}^2 V_{oh}(V_p^2 - V_n^2)}$	$\frac{k' V_p^2(V_{oh} + V_n)^2}{R_{off}^2 V_{oh}(V_p + V_n)(2V_{oh} - V_p + V_n)}$
Максимальная частота	$\frac{k' \min(V_n^2, (V_{oh} - V_p)^2)}{2R_{on}^2(V_p + V_n)}$	$\frac{k' V_n^2(V_{oh} - V_p)^2}{R_{on}^2 V_{oh}(V_p^2 - V_n^2)}$	$\frac{k' V_n^2(V_{oh} - V_p)^2}{R_{on}^2 V_{oh}(V_p + V_n)(2V_{oh} - V_p + V_n)}$
Условия генерации 1	$R_{on,i} < R_{m,i} < R_{off,i}$		
Условия генерации 2	$(-1)^l V_p V_{ol} > (-1)^l V_n V_{oh}$		

**l* = 0 для позитивного компаратора, *l* = 1 для негативного компаратора.

Отметим, что для рассматриваемого варианта напряжение на мемристоре $v = V_i$. Подставляя (9) и (11) в (10), получим дифференциальное уравнение первого порядка с разделяющимися переменными. Его решения позволяют определить все характеристики процесса генерации, в том числе временную диаграмму: форму токов, напряжений и изменение сопротивления во времени (рис. 3).

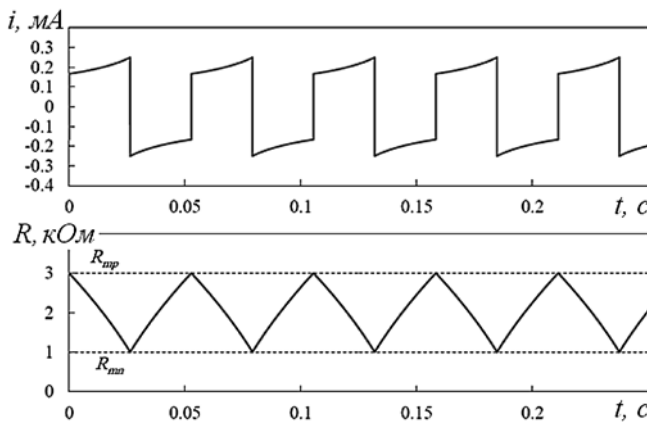


Рис. 3. Временные диаграммы позитивного *R—M*-генератора [9]

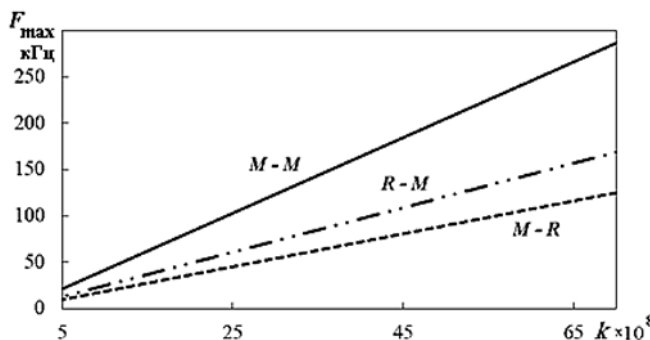


Рис. 4. Зависимость максимальной частоты генерации от параметров [9]

Приведенная временная диаграмма получена при моделировании мемристорного генератора, изображенного на рис. 2, *в*, с параметрами: $R_{on} = 100$ Ом, $R_{off} = 38$ кОм, $k_R = 3,79 \cdot 10^8$, $R_a = 3$ кОм, $V_{oh} = 1$ В, $V_{ol} = -1$ В, $V_p = 0,75$ В, $V_n = -0,5$ В. Из рис. 3 видно, что при непрерывном изменении сопротивления мемристора ток в интегрирующей цепи мемристорного генератора изменяется скачками, это отличает его от генератора на триггере Шмитта.

Результаты анализа условий генерации и параметры генераторов перечисленных типов сведены в таблицу.

Представляет интерес зависимость максимально возможной частоты генерации различных типов мемристорных генераторов от коэффициента инерционности (рис. 4). Как следует из таблицы, минимальная частота генерации может быть на несколько порядков ниже. Различие в частотах у разных типов генераторов вполне объяснимо: у *M—M*-генераторов одновременно изменяются в разных направлениях сопротивления двух мемристоров, что повышает скорость переходных процессов, а у *M—R*-генераторов интегрирующий элемент находится в цепи обратной связи и скорость переходного процесса падает.

Проведенные испытания экспериментальных образцов подобных генераторов подтвердили полученные теоретические оценки и расчеты. В целом можно сделать вывод, что подобные мемристорные генераторы могут оказаться проще, чем генераторы, использующие конденсаторы. Это особенно важно в области низких частот, где требуются конденсаторы большой емкости и, соответственно, больших размеров.

3.2. Безиндуктивные гармонические мемристорные генераторы

В рассмотренных выше системах первого порядка при линейно-кусочных характеристиках

элементов имеют место типичные релаксационные колебания.

Для перехода к гармоническим и квазигармоническим колебаниям в генераторы необходимо вводить дополнительные элементы. Введение реактивных элементов для повышения порядка систем является простейшим способом такого дополнения. В результате система начинает обладать частотными характеристиками с фильтрующими свойствами, позволяющими выделять гармонические составляющие. По такому принципу построен ряд гармонических *RC*-генераторов.

Генератор Вина представляет собой операционный усилитель, охваченный частотно-зависимой положительной обратной связью через мост Вина. Мост содержит четыре резистора и два конденсатора [10]. Генератор Вина может формировать гармонический сигнал в широком диапазоне частот. На рис. 5, *a* приведен вариант генератора Вина, у которого резистор в цепи обратной связи заменен на мемристор.

Условие генерации в таком генераторе можно записать в виде

$$1 + \frac{R_{Ma}}{R_2} = \frac{R_3}{R_4}, \quad (12)$$

где R_{Ma} — усредненное значение сопротивления мемристора за период колебаний. Частота генерации в этом случае определяется следующим образом:

$$F = \frac{1}{2\pi C \sqrt{R_{Ma} R_2}}. \quad (13)$$

Было выполнено моделирование генератора Вина с параметрами $R_2 = 5$ кОм, $C = 3,2$ мкФ, значение начального мемристанса выбирали в диапазоне 4,1...5,9 кОм, а отношение сопротивлений R_3/R_4 — таким, чтобы выполнялось условие генерации (12). При этом наблюдались квазисинусоидальные колебания как мемристанса, так и выходного сигнала. Как следовало ожидать, колебания и по амплитуде, и по частоте зависят от начального значения мемристанса, сама частота в процессе генерации оказывается промодулированной. Таким образом, встраивание мемристора в генератор Вина сохраняет свойства генерации, по крайней мере, при условии малых изменений мемристанса.

Другим примером мемристорного *RC*-генератора является фазосдвигающий генератор [11] (рис. 5, *б*). Принцип его работы заключается во введении отрицательной обратной связи из трех *RC*-цепочек, обеспечивающих общий сдвиг гармонического сигнала на 180° . Замена резисторов на мемристоры, как и в предыдущем случае, сохраняет работоспособность генератора при условии малых изменений

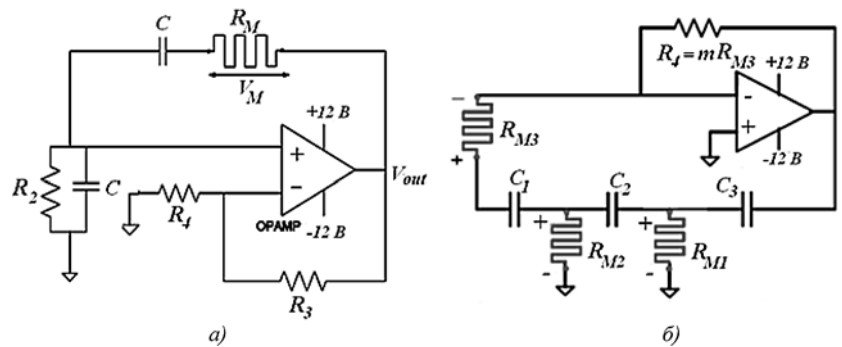


Рис. 5. *RC*-генераторы с мемристором: *a* — генератор Вина [10]; *б* — фазосдвигающий генератор [11]

мемристанса. Вместе с тем появляется зависимость параметров генерации от их начальных значений.

4. Мемристорные хаотические генераторы

Рассмотренные выше типы сравнительно простых генераторов решают частные вопросы. Перспективы подобных генераторов определяются их реальной конкурентоспособностью, а она в значительной мере зависит от технологичности самих мемристоров.

Мемристорные генераторы с колебательными контурами привлекают в настоящее время большее внимание. Причина состоит в том, что симбиоз линейной колебательной системы с нелинейными элементами типа мемристоров открывает возможности создания генераторов с уникальными характеристиками. Это обусловлено наличием гистерезиса в ВАХ мемристора. Гистерезис существенно усложняет фазовый портрет генератора с возможным превращением последнего в генератор хаотических колебаний. Подобные хаотические генераторы (ХГ) вызывают повышенный интерес, так как могут найти применение в ряде важных приложений, в том числе в криптографии, в нейроморфных системах и др.

В 80-е годы прошлого века была показана возможность хаотической генерации с использованием диодов Чуа — двухполюсников, имеющих участок с отрицательным дифференциальным сопротивлением (ОДС) определенного вида [20]. Диоды Чуа могут быть реализованы различным образом, в том числе путем сочетания мемристоров с ОДС приборами. За последние годы выполнены всесторонний анализ и моделирование различных мемристорных хаотических генераторов (МХГ) [21]. Для них разработаны специальные математические модели мемристоров [22] и для исследований МХГ создана платформы [23].

Известно, что для появления хаотического поведения динамической системы в ее составе необходимы, по крайней мере, один нелинейный элемент, один элемент с отрицательным дифференциальным сопротивлением, три накопителя энергии.

Каноническая модель МХГ обладает этими качествами [7]. Ее можно представить в виде эквивалентной схемы (рис. 6), содержащей индуктивность, два конденсатора, резистор (второй резистор в простом генераторе может отсутствовать) и диод Чуа, скомбинированный из элемента с ОДС и мемристора. В данном случае модель мемристора с управляющим напряжением описывается следующими соотношениями:

$$\frac{dw}{dt} = v; \quad (14)$$

$$i = G(w)v = \begin{cases} av, & \text{если } |w| < 1, \\ bv, & \text{если } |w| \geq 1, \end{cases} \quad (15)$$

где w — переменная, описывающая состояние мемристора. В отличие от линейно-дрейфовой модели проводимость мемристора в данном случае является немонотонной и разрывной функцией от интеграла напряжения. Эта модель в зависимости от параметров элементов реализует богатый набор динамических состояний и эффектов: потерю устойчивости, мягкое и жесткое возбуждение автоколебаний, бифуркации периодических движений, сопровождающиеся переходом к хаотическим колебаниям, сильную зависимость поведения от начальных состояний системы, гистерезис.

Проверка хаотичности носит в основном качественный характер. Критериями наиболее часто выступают: особенности фазового портрета системы и характер отображения Пуанкаре; широкополос-

ный низкочастотный шум и дельтаобразная автокорреляционная функция в начале координат; бифуркационные диаграммы; положительный показатель Ляпунова на траектории аттрактора.

Поведение схемы на рис. 6, а описывается следующей системой уравнений:

$$C_1 \frac{dv_1}{dt} = \frac{v_2 - v_1}{R} + Gv_1 - G(w)v_1; \quad (16)$$

$$C_2 \frac{dv_2}{dt} = \frac{v_1 - v_2}{R} - i; \quad (17)$$

$$L \frac{di}{dt} = v_2 - ri; \quad (18)$$

$$\frac{dw}{dt} = v_2. \quad (19)$$

Компьютерное моделирование показывает, что система уравнений (16)–(19) при параметрах $a = 0,3$, $b = 0,8$, $G = 1,5$ (0,6), $r = 0,027$, $C_1 = 0,1$, $C_2 = 1$, $L = 0,077$ имеет хаотический аттрактор (рис. 6, б). Вычисление показателя Ляпунова по выборке значений переменных дает положительное значение, примерно равное 0,0779.

При упрощении генератора изменяется и характер генерации. При исключении из схемы на рис. 6 линейных сопротивлений она превращается в мемристорный генератор Ван-дер-Поля, который вследствие гистерезиса мемристора может иметь два предельных цикла. При последующем упрощении схемы генерируемые колебания становятся релаксационными при исключении конденсатора или исчезают при исключении индуктивности.

В приведенном варианте хаотического генератора использована модель мемристора с двумя значениями проводимости. Проведены исследования генераторов и с другими моделями мемристоров, которые также обладают хаотической генерацией. Следует отметить, что многие из них не соответствуют реальным мемристорам.

При усложнении рассматриваемого генератора возможен переход к еще более хаотическому поведению, а именно, к гиперхаосу. Гиперхаос характеризуется наличием двух и более положительных показателей Ляпунова. Показано, что введением в канонический МХГ дополнительной цепи (контура) с индуктивностью удается получить гиперхаотический режим. Другой вариант — два связанных МХГ, они также могут иметь подобный режим.

Простейший МХГ — одноконтурная цепь, содержащая два линейных реактивных элемента и мемристор [13]

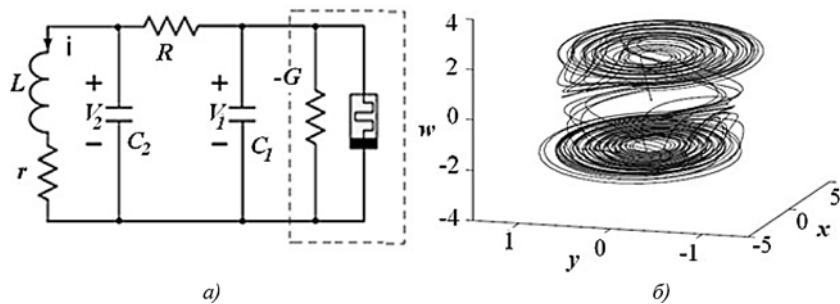


Рис. 6. Канонический МХГ: а — электрическая схема; б — фазовый портрет [7]

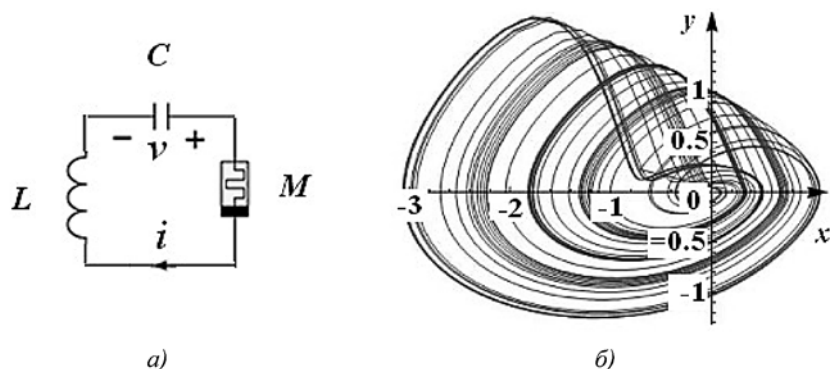


Рис. 7. Простейший МХГ: а — схема; б — фазовый портрет МХГ [13]

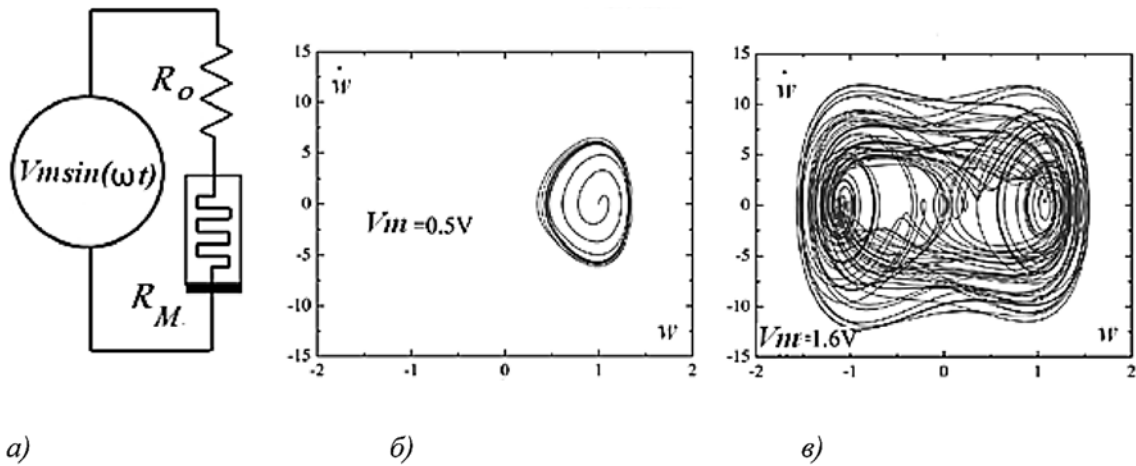


Рис. 8. МХГ с внешним возбуждением:

а — схема; б — устойчивая траектория при слабом возбуждении; в — странный аттрактор при сильном возбуждении [14]

(рис. 7). Поскольку по законам Кирхгофа ток во всех элементах контура одинаков и напряжения линейно зависимы, то состояние мемристора не является независимым. Система, содержащая три инерционных элемента, становится системой второго порядка. Заменой мемристора на мемристорную систему и введением нелинейного мемристанса в сочетании с усложнением формулы управляющей характеристики удастся добавить дополнительную степень свободы и добиться хаотичности. Отметим, что у мемристора появляется участок с ОДС. Поведение в этом случае описывается системой трех независимых дифференциальных уравнений

$$C \frac{dv}{dt} = i,$$

$$L \frac{di}{dt} = -[v + \beta(w^2 - 1)],$$

$$\frac{dw}{dt} = -y - \alpha w + iw.$$

При значениях параметров $C = 1$, $L = 3$, $\beta = 1,5$, $\alpha = 0,6$ при начальных условиях $\{0,1, 0, 0,1\}$ схема возбуждается и начинается хаотическая генерация (рис. 7, в).

Таким образом, для генератора хаоса достаточно двух реактивных линейных элементов и нелинейного интегрирующего элемента с ОДС. Для реализации подобного генератора целесообразно вместо индуктивности использовать гиратор.

Предложен и принципиально отличный подход — исключение колебательного контура при усложнении управляющей характеристики мемристора [14]. Ее изменение записывается через дифференциальное уравнение второго порядка, что в нашем представлении выглядит следующим образом:

$$\frac{d^2 w}{dt^2} = -c_1 \frac{dw}{dt} - c_2 w(w^2 - 1) + c_3 v.$$

Мемристор, управляемый в данном случае напряжением, подключается через резистор к источнику гармонического сигнала (рис. 8, а).

При отсутствии внешних воздействий УХ такой системы имеет равновесные устойчивые точки 1 и -1 . Если вывести систему из равновесия малой амплитудой возбуждающего сигнала, то после его прекращения система возвращается в устойчивое состояние (рис. 8, б). Однако при воздействиях сигнала с большой амплитудой происходит самовозбуждение, и генерация носит хаотический характер (рис. 8, в), т. е. имеет место "жесткая" хаотическая генерация.

При теоретических исследованиях МХГ, как правило, используются симметричные функции мемристанса (мемкондактанса), и допускаются отрицательные значения состояния мемристора. У реальных приборов сопротивления положительны и функция асимметрична. Это затрудняет перенос теоретических результатов на реализуемые схемы. У мемристоров с несимметричными характеристиками также наблюдали процессы ХГ, но область их существования имеет более сложный вид в пространстве параметров. Поэтому представляет интерес симметризация характеристик мемристоров.

Одним из способов симметризации является антипараллельное включение двух одинаковых мемристоров [15] (рис. 9). В этом случае суммарная

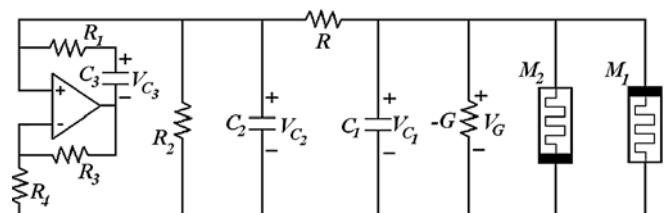


Рис. 9. Безиндуктивная схема МХГ с антипараллельными мемристорами [15]

управляющая характеристика становится симметричной, по крайней мере, при низких скоростях изменения состояния, что имеет место в МХГ [15]. Кроме того, у такого варианта генератора ослаблена зависимость характера генерации от неизбежного разброса параметров реальных приборов. Существенной особенностью МХГ (рис. 9) является замена индуктивности гиратором, на основе операционного усилителя с мостом Вина. Это позволяет упростить реализацию подобного генератора. Однако наличие в его составе конденсаторов с достаточно большой емкостью (порядка десятков микрофарад), создает трудности для его реализации по современной интегральной технологии.

Заключение

Нелинейные и инерционные свойства мемристора открывают новые возможности реализации нелинейных схем со сложным поведением. В статье показаны особенности разработки генераторов на основе мемристорных элементов. Представлен новый класс безреактивных автогенераторов. Обсуждаются перестраиваемые R — C -генераторы и генераторы хаотических сигналов.

Список литературы

1. Chua L. Memristor—Missing Circuit Element // IEEE Trans. Circuit Theory. 1971. Vol. 18, N. 9. P. 507—519.
2. Chua L., Kang S. M. Memristive Devices and Systems // Proc. IEEE. 1976. Vol. 64, N. 2. P. 209—223.
3. Strukov D. B., Snider G. S., Stewart D. R. The Missing Memristor Found // Nature. 2008. Vol. 453. P. 80—83.
4. Mazumder P., Kang S. M., Waser R. Memristors: Devices, Models and Applications // Proc. IEEE. 2012. Vol. 100, N. 6. P. 1911—1919.
5. Mickel P. R., Lohn A. J., Marinella M. J. Memristive Switching: Physical Mechanisms and Applications // Modern Physics Letters B. 2014. Vol. 28, N. 10, 1430003.
6. Letellier C., Ginoux J. M. Development of the Nonlinear Dynamical Systems Theory From Radio Engineering to Electronics // Int. J. Bifurcation Chaos. Vol. 19, N. 07. P. 2131—2163.
7. Itoh M., Chua L. O. Memristor Oscillators // Int. J. Bifurcation Chaos. 2008. Vol. 18, N. 11. P. 3183—3206.
8. Zidan M., Omran H., Radwan A. G., Salama K. N. Memristor-Based Reactance-Less Oscillator // Electronics Letters. 2011. Vol. 47, N. 22. P. 1220—1221.
9. Zidan M., Omran H., Smith C., Radwan A. G., Salama K. N. A Family of Memristor Based Reactance Less Oscillators // Int. J. Circuit Theory and Applications. 2014. Vol. 42, N. 11. P. 1103—1122.
10. Talukdar A., Radwan A. G., Salama K. N. Generalized Model for Memristor Based Wien Family Oscillators // Microelectronics Journal, 2011. Vol. 42. N. 9. P. 1032—1038.
11. Talukdar A., Radwan A. G., Salama K. N. Non Linear Dynamics of Memristor Based 3 Order Oscillatory System // Microelectronics journal. 2012. Vol. 43, N. 3. P. 169—175.
12. Bahgat A., Salama K. N. Memristor-based mono-stable oscillator // Arxiv preprint. 2012. Arxiv 1207.0847.
13. Muthuswamy B., Chua L. O. Simplest Chaotic Circuit // Int. J. Bifurcation and Chaos. 2010. Vol. 20, N. 5. P. 1567—1580.
14. Driscoll T., Pershin Y., Basov D., Di Ventra M. Chaotic Memristor // Appl. Phys A. 2011. Vol. 102. P. 885—889.
15. Buscarino A. A Gallery of Chaotic Oscillators Based on HP Memristor // Int. J. Bifurcation Chaos. 2013. Vol. 23, N. 5, 1330015.
16. Pabst O., Schmidt T. Frequency Dependent Rectifier Memristor Bridge Used as a Programmable Synaptic Membrane Voltage Generator // J. Electr. Bioimp. 2013. Vol. 4. P. 23—32.
17. Serrano-Gotarredona T., Masquelier T., Prodromakis T. et al. STD Pand STDP Variations with Memristors for Spiking Neuro-morphic Learning Systems // Frontiers in Neuroscience. 2013. Vol. 7. P. 1—15.
18. Corinto A., Ascoli A., Gilliet M. et al. Anaiysis of Current-Voltage Characteristics for Memristive Elements in Pattern Recognition Systems // Journal of Circuit Theory and Applications. 2012. Vol. 40, N. 12. P. 1277—1320.
19. Ракитин В. В., Русаков С. Г. Электрические модели мемристорных элементов // Информационные технологии. 2014. № 8. С. 50—58.
20. Matsumoto T., Chua L. O., Komuro M. The Double Scroll // IEEE Trans. on Circuits and Systems, 1985. Vol. 32. P. 797—818.
21. Kennedy P. Genealogy of Chua's Circuit // Chaos, Cnn, Memristors and Beyond: P. 3—24.
22. Wang L., Drakakis E., Duan S. K. et al. Memristor Model and its Application for Chaos Generation // Int. J. Bifurcation Chaos. 2012. Vol. 22, N. 06. 1250133.
23. Arik S. Reconfigurable Hardware Platform for Experimental Testing and Verifying of Memristor-Based Chaotic Systems // Int. J. Bifurcation Chaos. 2014. Vol. 23, N. 10. 1450145.

V. V. Rakitin, Ph. D., Leading Researcher; vlarak@rambler.ru,

S. G. Rusakov, D. Sc., Corresponding member of RAS, Chief Researcher; rusakov@ippm.ru,
Institute for Design Problems in Microelectronics of Russian Academy of Sciences (IPPM RAS), Moscow

Memristor-Based Oscillator Circuits

The current status of the memristor-based oscillator development is considered. The memristor oscillators are represented a new class of nonlinear circuits. The nonlinear and inertial properties of memristor devices open the new opportunities for design of nonlinear circuits in different applications. The underlying principles of oscillator design are given with using the basic memristor property of analog memory. The features to construct the self-oscillator circuits are shown that are based on exploitation of memristor characteristics. The basic circuit solutions of inductance-less oscillators and reactance-less oscillators are discussed. The advantages of applying the memristors for chaotic oscillators design are considered.

Keywords: memristors, memristive systems, memristor oscillator, inductance-less oscillators, reactance-less oscillators, chaotic circuits

References

1. Chua L. O. Memristor—Missing Circuit Element, *IEEE Trans. Circuit Theory*, 1971, vol. 18, no. 9, pp. 507—519.
2. Chua L. O., Kang S. M. Memristive Devices and Systems, *Proc. IEEE*, 1976, vol. 64, no. 2, pp. 209—223.
3. Strukov D. B., Snider G. S., Stewart D. R. The Missing Memristor Found, *Nature*, 2008, vol. 453, pp. 80—83.
4. Mazumder P., Kang S. M., Waser R. Memristors: Devices, Models and Applications, *Proc. IEEE*, 2012, vol. 100, no. 6, pp. 1911—1919.
5. Mickel P. R., Lohn A. J., Marinella M. J. Memristive Switching: Physical Mechanisms and Applications, *Modern Physics Letters B*, 2014, vol. 28, no. 10, 1430003.
6. Letellier C., Ginoux J. M. Development of the Nonlinear Dynamical Systems Theory from Radio Engineering to Electronics, *Int. J. Bifurcation Chaos*, 2009, vol. 19, no. 07, pp. 2131—2163.
7. Itoh M., Chua L. O. Memristor Oscillators, *Int. J. Bifurcation Chaos*, 2008, vol. 18, no. 11, pp. 3183—3206.
8. Zidan M., Omran H., Radwan A. G., Salama K. N. Memristor-Based Reactance-Less Oscillator, *Electronics Letters*, 2011, vol. 47, no. 22, pp. 1220—1221.
9. Zidan M., Omran H., Smith C., Radwan A. G., Salama K. N. A Family of Memristor Based Reactance Less Oscillators, *Int. J. Circuit Theory and Applications*, 2014, vol. 42, no. 11, pp. 1103—1122.
10. Talukdar A., Radwan A. G., Salama K. N. Generalized Model for Memristor Based Wien Family Oscillator, *Microelectronic Journal*, 2011, vol. 42, no. 9, pp. 1032—1038.
11. Talukdar A., Radwan A. G., Salama K. N. Non Linear Dynamics of Memristor Based 3 Order Oscillatory System, *Microelectronics Journal*, 2012, vol. 43, no. 3, pp. 169—175.
12. Bahgat A., Salama K. N. Memristor-based mono-stable oscillator, Arxiv preprint, 2012, Arxiv, 1207.0847.
13. Muthuswamy B., Chua L. O. Simplest Chaotic Circuit, *Int. J. Bifurcation and Chaos*, 2010, vol. 20, no. 5, pp. 1567—1580.
14. Driscoll T., Pershin Y., Basov D., Di Ventra M. Chaotic Memristor, *Appl. Phys.*, 2011, vol. 102, no. 4, pp. 885—889.
15. Buscarino A. A. Gallery of Chaotic Oscillators Based on HP Memristor, *Int. J. Bifurcation Chaos*, 2013, vol. 23, no. 5, 1330015.
16. Pabst O., Schmidt T. Frequency Dependent Rectifier Memristor Bridge Used as a Programmable Synaptic Membrane Voltage Generator, *J. Electr. Bioimp.*, 2013, vol. 4, pp. 23—32.
17. Serrano-Gotarredona T., Masquelier T., Prodromakis T., Indiveri G., Linares-Barranco B. STDP and STDP Variations with Memristors for Spiking Neuromorphic Learning Systems, *Frontiers in Neuroscience*. 2013, vol. 7, pp. 1—15.
18. Corinto A., Ascoli A., Gilliet M. Analysis of Current-Voltage Characteristics for Memristive Elements in Pattern Recognition Systems, *Journal of Circuit Theory and Applications*, 2012, vol. 40, no. 12, pp. 1277—1320.
19. Rakitin V. V., Rusakov S. G. Electrical Models of Memristor Elements, *Information Technologies*, 2014, no. 8, pp. 50—58 (in Russian).
20. Matsumoto T., Chua L. O., Komuro M. The Double Scroll, *IEEE Trans. on Circuits and Systems*, 1985, vol. 32, no. 8, pp. 798—818.
21. Kennedy P. Genealogy of Chua's Circuit, *Chaos, Cnn, Memristors and Beyond*, pp. 3—24.
22. Wang L. D., Drakakis E., Duan S. K., He P. F., Liao X. F. Memristor Model and its Application for Chaos Generation, *Int. J. Bifurcation Chaos*, 2012, vol. 22, no. 06, 1250133.
23. Arik S. Reconfigurable Hardware Platform for Experimental Testing and Verifying of Memristor-Based Chaotic Systems, *Int. J. Bifurcation Chaos*, 2014, vol. 23, no. 10, 1450145.

ИНФОРМАЦИЯ

Инфокоммуникационные технологии и общество

Многочисленные социологические теории, в которых осмысливается состояние человеческой цивилизации в начале третьего тысячелетия, практически едины в характеристике современного общества как информационного, т.е. основанного на всепроникающем применении компьютеров и инфокоммуникационных технологий (ИКТ). И именно в сегодняшнем стремительно меняющемся мире особенно важна не простая констатация, но и глубокое понимание самой тесной взаимосвязи технического прогресса с развитием социума — хотя нельзя не сказать, что у профессионалов в компьютерной области такое понимание сложилось достаточно давно. Так, одно из старейших и влиятельнейших профессиональных обществ — IFIP (International Federation of Information Processing) вот уже сорок лет проводит конференции *Human Choice and Computers*. Очередная конференция *HCC11* прошла с 30 июля по 1 августа 2014 г. в университете г. Турку (Финляндия).

Доклады, включенные в программу конференции, охватывали широкий спектр вопросов: этика компьютеринга, ИКТ и трудовые отношения, ИКТ и устойчивое развитие, ИКТ и право, виртуальность и реальность, социальная ответственность работников в сфере ИКТ, роль ИКТ в развивающихся странах и др.

По традиции широко были представлены работы исторической направленности. В основном на впервые вводимых в научный оборот документах докладе Мари д'Юдекем-Живерс (университет г. Намюра, Бельгия) "Математическая машина IRSIA-FRNS" была рассказана история создания в 1950-х годах первого бельгийского электронного компьютера, а Кристофер Лесли и Патрик Гричка из Нью-Йоркского университета в своем выступлении охарактеризовали особенности (вероятно, общие для всех восточно-европейских стран) формирования компьютерных сетей в Польше и их использования в борьбе против тоталитарного режима.

В двух докладах в исторической ретроспективе были рассмотрены вопросы внедрения информационных технологий в образование. Рамон Пуоигьянер (университет Балеарских островов) и Хорди Форнес (Барселонский технический университет) основывались на опыте испанского высшего образования в целом, а Артур Татнал (университет Мельбурна) и Билл Дэви (университет РМИТ, Мельбурн) — на опыте университетов и школ австралийского штата Виктория.

Приятно, что активное участие в работе исторической секции приняли российские специалисты. Марина Смолевицкая (Политехнический музей, Москва) описала хранящиеся в Политехническом музее личные фонды основоположников советской вычислительной техники С. А. Лебедева, В. С. Бурцева, И. С. Брука и М. А. Карцева. Пожалуй, самую оживленную дискуссию вызвал доклад Валерия Шилова и Сергея Силантьева (Москва) "Разум и ортодоксия, или Уроки судьбы русской логической машины", в котором авторы на примере судьбы выдающегося российского ученого Александра Николаевича Шукарева (1864—1936) наглядно показали пагубность идеологического вмешательства государства в научное творчество.

Остается добавить, что издательством *Springer* в известной серии *IFIP Advances in Information and Communication Technology* уже к началу конференции был издан сборник трудов (ICT and Society. Eds.: K. Kimppa, D. Whitehouse et al. Springer, 2014).

В. В. Шилов,

Национальный исследовательский университет "Высшая школа экономики"

С. А. Силантьев

МАТИ — Российский государственный технологический университет имени К. Э. Циолковского

ЖУРНАЛ В ЖУРНАЛЕ



**НЕЙРОСЕТЕВЫЕ
ТЕХНОЛОГИИ**

№ 10
ОКТАБРЬ
2015

Главный редактор:

ГАЛУШКИН А.И.

Редакционная коллегия:

АВЕДЬЯН Э.Д.
БАЗИЯН Б.Х.
БЕНЕВОЛЕНСКИЙ С.Б.
БОРИСОВ В.В.
ГОРБАЧЕНКО В.И.
ЖДАНОВ А.А.
ЗЕФИРОВ Н.С.
ЗОЗУЛЯ Ю.И.
КРИЖИЖАНОВСКИЙ Б.В.
КУДРЯВЦЕВ В.Б.
КУЛИК С.Д.
КУРАВСКИЙ Л.С.
РЕДЬКО В.Г.
РУДИНСКИЙ А.В.
СИМОРОВ С.Н.
ФЕДУЛОВ А.С.
ЧЕРВЯКОВ Н.И.

**Иностранные
члены редколлегии:**

БОЯНОВ К.
ВЕЛИЧКОВСКИЙ Б. М.
ГРАБАРЧУК В.
РУТКОВСКИЙ Л.

Редакция:

БЕЗМЕНОВА М.Ю.
ГРИГОРИН-РЯБОВА Е.В.
ЛЫСЕНКО А.В.
ЧУГУНОВА А.В.

Барский А. Б.

Обучаемые и самообучающиеся системы распознавания,
управления и принятия решений на логических нейронных
сетях 783

Куравский Л. С., Мармалюк П. А., Юрьев Г. А., Думин П. Н.

Методы численной идентификации марковских моделей
и их сравнительный анализ 792

А. Б. Барский, д-р техн. наук, проф., e-mail: arkbarsk@mail.ru,
Московский государственный университет путей сообщения

Обучаемые и самообучающиеся системы распознавания, управления и принятия решений на логических нейронных сетях

Рассматривается применение логических нейронных сетей в ассоциативных "бесформульных" вычислениях трудно-формализуемых задач и задач имитационного моделирования, в медицинских информационно-справочных системах, в системах идентификации пользователя, в адаптивной пошаговой маршрутизации беспроводных телекоммуникационных и транспортных сетей, а также в самообучающихся системах управления по нечетким данным.

Приводятся рекомендации по более качественному воплощению некоторых известных систем принятия решений.

Ключевые слова: логическая нейронная сеть, функция активации, матрица следования, ассоциативные вычисления, информационно-справочные системы, самообучающиеся системы управления

Введение

В основе логических нейронных сетей лежит математическая логика событий, объединяющая начальные сведения из математической логики и теории вероятностей. Возбуждение рецепторов формируется на основе достоверности высказываний о принадлежности данных. Это обеспечивает единую систему измерения исходных данных и резко расширяет область применения аппарата нейронных сетей для построения практически любой системы принятия решений, для данных любой природы, типов и формы представления. Возбуждение нейронов выходного слоя указывает на принимаемое решение в форме значения, текста, управляющего сигнала и пр. Сети сводятся к однослойным. В них обучающие ситуации непосредственно связываются с соответствующими решениями. Логические нейронные сети отличаются невысокими требованиями к квалификации разработчика, несложными расчетами, высоким быстродействием и возможностью распараллеливания.

Реализация численного метода ассоциативных вычислений с помощью логической нейронной сети

Пусть система принятия решений описывается набором логических выражений вида

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &\rightarrow R_1; \\ &\dots\dots\dots \\ f_m(x_1, x_2, \dots, x_n) &\rightarrow R_m. \end{aligned} \quad (1)$$

Булевы переменные x_i ($i = 1, \dots, n$) фиксируют наличие или отсутствие *событий* i , отображенных высказываниями вида: "поезд приходит в 20.00", "головная боль наблюдается", "X принимает значение из диапазона 900... 1100" и т. д. Комбинация значений этих переменных (точка) характеризует *ситуацию на факторном пространстве событий*.

Определенные на этом пространстве функции (1), в левой части представляющие композицию опе-

раций конъюнкции (И) и дизъюнкции (ИЛИ), утверждают причинно-следственную связь. Если функция f_j от значения факторов данной ситуации принимает значение ИСТИНА, то принимается решение R_j . Оно может быть высказыванием самого разного содержания: от численного или векторного значения функции многих переменных до сложнейших текстов рекомендательного характера (алгоритма действий, команд запуска программ, имитирующих движение реагирующего объекта, оптимальной стратегии управления).

Система корректна, если выполняются условия полноты, однозначности и непротиворечивости предлагаемых решений.

Включение исчерпывающих множеств событий по каждому фактору приводит к отсутствию отрицаний в рассматриваемых логических выражениях. Возможность формирования по каждой логической функции ее дизъюнктивной нормальной формы, с учетом закрепления одного и того же решения за разными конъюнкциями, позволяет получить логические цепочки единичной длины. Это, в свою очередь, приводит к преимущественному и достаточному изучению *однослойных логических нейронных сетей* [1–4], где булевы переменные заменяются действительными на всем диапазоне значений от нуля до единицы. Введенные переменные интерпретируются как *достоверность высказываний о принадлежности данных*. Однослойные логические нейронные сети содержат рецепторный и выходной слои. Нейроны выходного слоя реализуют простейшую пороговую функцию активации, примеры которых будут рассмотрены. Связи, ведущие от рецепторов к нейронам выходного слоя, могут обладать весами в диапазоне [0, 1].

Фактически логическая нейронная сеть на основе опытных или расчетных данных реализует таблицу вида "если {ситуация}, то — {решение}", вполне соответствующую идеям логического вывода и ситуационного управления Д. А. Поспелова [5, 6].

Для обработки "промежуточных" ситуаций, возникающих в рамках запроса, должна быть обоснована процедура интерполяции. Эта процедура выполняется в результате задания возникшей "промежуточной" ситуации на рецепторах, счета функции активации нейронов и определения решения, соответствующего наиболее возбуждавшемуся нейрону. Если решения обладают численными значениями, то по значениям возбуждения нескольких нейронов может находиться среднее.

Аппроксимация опыта и ассоциативные "бесформульные" вычисления

Давно известно, что исследователи, использующие современный метод имитационного моделирования сложных систем, прекратили попытки построения функций аппроксимации для графического представления своих результатов. Большое число формально несовместимых факторов приводило бы к формированию большого числа двух- и трехмерных таблиц для выделенных переменных. Эти таблицы составлялись бы для фиксированных значений других переменных, рассматриваемых в качестве параметров, образующих множество точек в многомерном пространстве. Наглядность такого

представления при желании получить огромное число графиков проблематична.

Исследователи формируют *базы знаний*, представляя свои результаты в виде таблиц, связывающих испытываемые точки факторного пространства (ситуации) с векторами найденных значений результатов — решений по данным ситуациям. Наиболее характерно это для полигонных испытаний, используемых для калибровки моделей. Таким образом, для дальнейших исследований предоставляется далеко не полная оценка результатов моделирования во всем факторном пространстве.

Чтобы превратить эти оценки в количественные на всем факторном пространстве при допущении об отсутствии особых точек, необходимо создать механизм расширения их на всю испытываемую область. Используется метод ассоциативного нахождения "промежуточных" решений, заключающийся в определении: "на какую (какие) из известных ситуаций и в какой мере похожа вновь возникшая ситуация и как по базе знаний выбрать или построить соответствующее ей решение". При этом по нескольким "похожим" ситуациям возможно усредненное решение.

Механизмом, представляющим собой численный метод "бесформульного" ассоциативного вывода, и

являются логические нейронные сети. С их помощью реализуются развиваемые информационно-справочные системы, объединяющие опыт и результаты моделирования в конкретной области исследований.

Например, в работе [7] исследуются результаты моделирования многосерверной базы данных с циркулирующими сегментами. Для множества точек факторного пространства — значений векторов $\{\lambda$ (интенсивность потока запросов), τ (длительность такта), $t_{\text{вып}}$ ("чистое" время выполнения запроса), n (число серверов), m (число циркулирующих сегментов)} — находятся значения вектора {среднее время выполнения запроса, вероятность выполнения}.

Информационной основой построения *обученной* нейронной сети (рис. 1) для информационно-справочной системы являются таблицы, построенные в процессе моделирования.

Значение V_i простейшей функции активации i -го нейрона находится в результате расчетов:

$$V_i := \begin{cases} 0, & \text{если } V \geq h; \\ 0, & \text{в противном случае,} \end{cases} \quad (2)$$

где $V = \frac{1}{k} \sum_{j=1}^k V_j$, k — число активных входов нейронов.

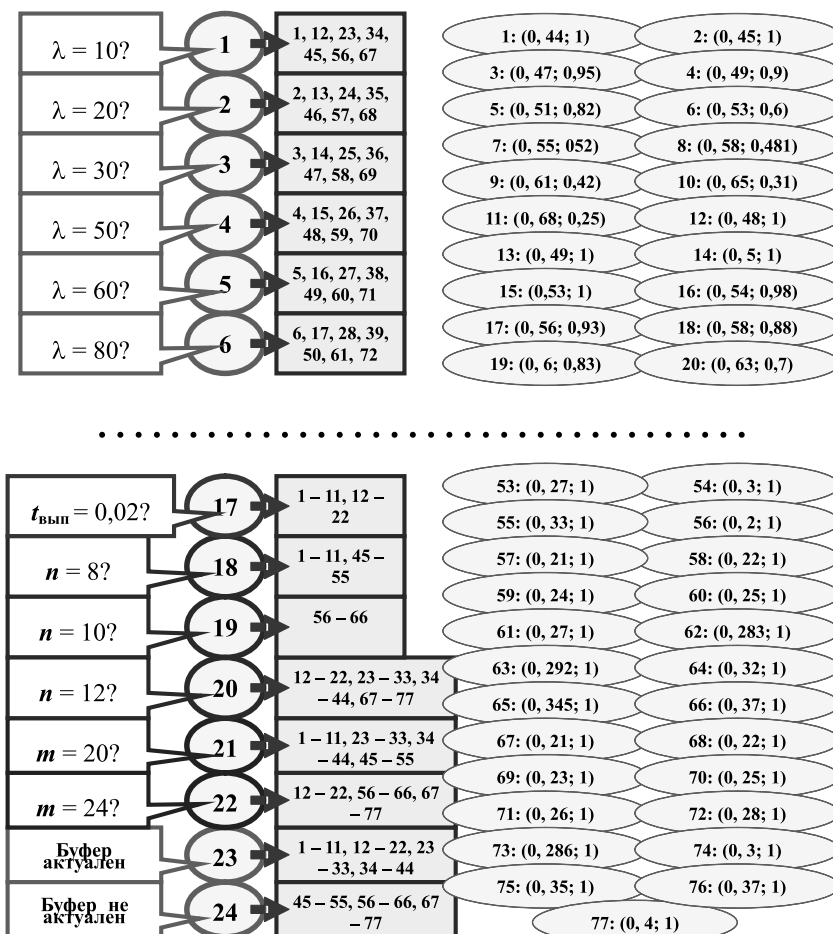


Рис. 1. Фрагмент информационно-справочной системы

Фрагмент матрицы следования для логической нейронной сети, реализующей информационно-справочную систему

Решения	Значения факторов															
	...	$\lambda = 140$	$\lambda = 160$	$\tau = 0,05$	$\tau = 0,1$	$\tau = 0,2$	$t_{\text{вып}} = 0,005$	$t_{\text{вып}} = 0,01$	$t_{\text{вып}} = 0,02$	$n = 8$	$n = 10$	$n = 12$	$m = 20$	$m = 24$	Буфер ограничен	Буфер не ограничен
Пример		0,7	0,3	0,5	0,5		0,6	0,4			0,5	0,5		1		1
10		1			1				1	1			1		1	
11			1		1				1	1			1		1	
32		1			1			1			1	1			1	
33			1		1			1			1	1			1	
76		1		1			1				1			1		1
77			1	1			1				1			1		1

Здесь V_j — величина возбуждения рецептора, подаваемая на вход нейрона. В рассматриваемом случае $k = 6$.

Максимально возбуждившийся нейрон указывает на решение. Решением является вектор, состоящий из двух компонент — времени $t_{\text{вып}}$ выполнения запроса и вероятности P его выполнения.

Направление синаптических связей на рис. 1 в прямоугольниках справа от рецепторов, а решения обозначены на "телах" нейронов (эллипсах) парой искомым чисел.

Поскольку исследования многопроцессорных баз данных далеки от завершения, веса всех синаптических связей на рис. 1 приняты равными единице. Пока мы не можем с достаточной уверенностью установить, в какой степени каждый фактор влияет на получаемый результат. Это может быть выявлено в течение длительного опыта эксплуатации.

Обработка нейросети основана на ее представлении в виде матрицы следования, фрагмент которой показан в табл. 1. Она легко обрабатывается с помощью EXEL.

Например, пусть известны значения факторов создающейся многосерверной базы данных с циркулирующими сегментами: $\lambda = 146$, $\tau = 0,075$, $t_{\text{вып}} = 0,007$, $n = 11$, $m = 12$, размер буфера не актуален (так как не ограничен) из-за значительного объема памяти сервера.

Для возбуждения рецепторов (в строке "Пример") для каждого фактора распределяем "единицу" обратно пропорционально расстоянию до двух "ближних" рецепторов: $V_{10} = 0,7$, $V_{11} = 0,3$, $V_{12} = V_{13} = 0,5$, $V_{15} = 0,6$, $V_{16} = 0,4$, $V_{19} = V_{20} = 0,5$, $V_{22} = 1$, $V_{24} = 1$.

Считаем функцию активации всех нейронов $1...77$ скалярным умножением каждой строки в матрице на строку "Пример". Полученные результаты делим на 6 (каждый нейрон имеет 6 входных связей) и сравниваем с порогом h .

В общем случае порог выбирается экспериментально, чтобы отсеять малые значения возбуждения

нейронов. В данном случае представляется целесообразным принять $h = 0,6$. Чтобы не загромождать пример, в табл. 1 отражены строки матрицы следования, соответствующие лишь некоторым нейронам, с ожидаемым существенным возбуждением.

Полученные значения возбуждения нейронов:

$$V_{10} = V_{11} = V_{32} = V_{33} = 0, V_{76} = 0,71, V_{77} = 0,65.$$

По другим строкам матрицы следования возбуждения равны нулю.

По известным формулам нахождения среднего и вероятности находим

$$t_{\text{вып}} = \frac{0,37 \cdot 0,71 + 0,4 \cdot 0,65}{0,71 + 0,65} = 0,38, P = 1.$$

Медицинские информационно-справочные системы

Если ассоциативные "бесформульные" вычисления допускают усреднение решения, то в медицинских системах диагностики и лечения результат должен быть вполне определенным — как руководство к действию. В большинстве случаев речь идет о получении четкого решения по нечетким данным. Это подтверждается практикой привлечения консилиума при решении спорных вопросов. Поэтому построение базы знаний на основе опыта высококвалифицированных специалистов и ее организация на базе механизма ассоциативного мышления, реализуемого логической нейронной сетью, является чрезвычайно важной задачей.

Исключение операции усреднения при использовании логической нейронной сети говорит о том, что единственный максимально возбуждившийся нейрон выходного слоя указывает на предлагаемое решение. Более того, это решение редко является численным — в этом случае усреднение было бы допустимо. В основном, решение указывает на стратегию или алгоритм лечения.

В работе [8] представлена логическая нейронная сеть, реализующая информационно-справочную систему для лечения хронических заболеваний почек. В основе этой системы лежат клинические практические рекомендации Международной Рабочей группы специалистов. Объем статьи не позволил целиком, в рамках одного рисунка, поместить сформированную логическую нейронную сеть. Да и матрица следования (табл. 2) как основной аппарат практических вычислений там не отражена.

Рассмотрим пример запроса. Пусть из-за неточного измерения скорости клубочковой фильтрации (СКФ)¹ с достоверностью 0,6 (по приблизительной оценке врача) установлено, что наблюдается 5-я стадия ХЗП — хронического заболевания почек. Предполагаемая достоверность 4-й стадии составляет 0,4. Возбуждается рецептор 6 сообщением ему значения 0,6. Рецептору 15 сообщается значение 0,4. Далее, пусть из-за неточного замера уровня фосфора в сыворотке крови рецептору 18 пришлось задать значение возбуждения 0,8, а рецептору 19 — значение 0,2. Аналогично, с учетом ошибок измерения интактного ПТГ (паратормона) формируются значения 0,2, 0,2, 0,6 возбуждения рецепторов 20, 21 и 22 соответственно. Запрос сформирован, возбуждение остальных рецепторов полагаем нулевым. При нулевом значении порога максимально, до величины 1,4, возбудится нейрон, соответствующий решению Y_{22} = (Поддерживать содержание фосфора сыворотки крови на уровне 2,7 мг/дл и выше, но не выше чем 4,6 мг/дл. Следует обеспечить терапию активными стероидами витамина D и т. д.)

Идентификация пользователя в компьютерной сети по "почерку"

Для примера выберем три характеристики почерка пользователя в системе "клиент — сервер", образующие его "портрет": скорость использования клавиш $a(key/s)$, частоту использования стрелки "←"

"Портреты" пользователей

Пользователь	Скорость ввода символов $a(key/s)$	Частота использования клавиши "←" $b(key/s)$	Частота использования клавиши "delete" $c(key/s)$
U1	3,84	0,21	0,51
U2	3,25	0,56	0,29
U3	4,26	0,46	0,18
U4	4,69	0,91	0,13
U5	4,31	0,82	0,55
U6	3,83	0,57	0,12
U7	3,84	0,72	0,77
U8	3,88	0,45	0,53
U9	3,82	0,53	0,15
U10	3,87	0,12	0,92

возврата $b(key/s)$, частоту использования клавиши "delete" $c(key/s)$.

В работе [9] на основе статистического анализа десяти пользователей получены оценки данных характеристик (табл. 3). Не подвергая их сомнению, рассматривая лишь в качестве примера, используем эти оценки, отображающие накопленный опыт, для построения логической нейронной сети. При этом предположим, что в результате длительных испытаний и эксплуатации установлено неодинаковое влияние выделенных характеристик пользователя на результат распознавания.

А именно, пусть в результате оценок оказалось целесообразным учитывать характеристику b с весом 0,8 (рис. 2, см. четвертую сторону обложки).

Выберем функцию f_i активации i -го нейрона, предварительно рассчитав значение

$$f = \frac{\sum_v \omega_v f_v}{\sum_v \omega_v}, \quad (3)$$

тогда $f_i = f$, если $f \geq h$, 0 — в противном случае.

Здесь ω_v — вес входной связи, параметр v "пробегает" по всем входным связям нейрона.

Таблица 4

Матрица следования логической нейронной сети распознавания пользователя

Пользователь	Рецепторы																													
	A1	A2	A3	A4	A5	A6	A7	A8	A9	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	
U1				1							0,8																			
U2	1														0,8										1					
U3							1						0,8										1							
U4									1										0,8		1									
U5								1										0,8												
U6		1														0,8				1										
U7			1														0,8													
U8				1								0,8														1				
U9		1												0,8								1								
U10					1					0,8																				1

¹ На рассматриваемом формальном уровне смысл специальных терминов раскрывать нецелесообразно.

Обозначим высказывания: $A1 = \langle a = 3,25 \rangle, \dots$
 $S10 = \langle c = 0,92 \rangle$. Матрица следования представлена в табл. 4.

Рассмотрим пример. Пусть сервер, периодически контролируя работающих пользователей, рассчитал вектор статистических характеристик $\{a = 3,4, b = 0,51, c = 0,3\}$. Порог распознавания $h = 0,6$.

Находятся два рецептора группы a , "близких" к значению 3,4, и "единица" делится между ними обратно пропорционально "расстоянию". В данном случае возбуждение $A1$ принимается равным 0,75, а $A2 = 0,25$. Аналогично находятся два рецептора, включающих значение $b = 0,54$, — $B5$ и $B6$. Тогда возбуждение первого полагается равным 0,67, а второго — 0,33. Аналогично, рецептор $S5$ возбуждается на величину 0,9, а рецептор $S6$ на величину 0,1.

С помощью функции активации найдем величины возбуждения нейронов выходного слоя. Максимальная из этих величин — величина возбуждения нейрона, указывающего на пользователя $U2$ и превысившая порог, равна 0,61.

Адаптивная пошаговая маршрутизация в беспроводной телекоммуникационной сети

Рассмотрим беспроводную телекоммуникационную систему, в которой компьютеры узлов составляют вычислительную сеть [10]. На этой сети задано *отношение смежности*. Информационные пакеты движутся к адресату по смежным компьютерам (узлам). Каждый маршрут не формируется весь сразу, а реализуется динамически, по шагам с учетом приоритетного обращения к смежным узлам (в зависимости от адреса назначения), текущей загрузки этих узлов и текущей оценки качества связи между узлами (в [10] не учитывается). Лишь в результате такого комплексного анализа может быть окончательно выбран узел смещения. Таким образом, одновременно реализуемые маршруты в сети оказывают взаимное влияние, и "разводить" их динамически надо так, чтобы избежать пиковых нагрузок узлов и в конечном итоге обеспечить максимальную пропускную способность сети, а также максимальную вероятность выполнения передачи.

На каждом i -м узле есть таблица T_i (табл. 5) предпочтительного смещения при передаче данных на все прочие узлы, кроме смежных.

Таблица 5

Таблица предпочтительного смещения из данного i -го узла для каждого (не смежного) адреса назначения

Узел (адрес) назначения	Вес смежного узла в направлении передачи			
A_1	ω_{11}	ω_{12}	ω_{1K}
.....
A_R	ω_{R1}	ω_{R2}	ω_{RK}

Здесь R — число узлов, в которые возможна передача пакетов из данного узла через один из смежных, K — число смежных узлов. (В общем случае K — переменная величина.)

При выборе весов ω учитывается территориальное взаимное расположение узлов. Так, очевидно, что приоритетной является та передача, при которой пакет приближается к узлу назначения, хотя в динамике загрузки сети может оказаться, что "круглой" путь ближе "прямого".

Предполагается, что компьютер каждого узла имеет буфер, в котором накапливаются пакеты для дальнейшей отправки. Перегрузка буферов (точнее, — достижение критической отметки) должна блокировать прием новых пакетов. Возможна блокировка передач по направлениям или в сети в целом. Потери информации не предполагаются, пользователь должен быть информирован об этой перегрузке для повторения запроса позже.

Управление передачей пакетов проводится с помощью логической нейронной сети, которая первоначально использует для каждого адреса предпочтительные направления передачи пакетов смежным узлам, найденные по табл. 4. Веса этих смещений используются в качестве весов синаптических связей. С помощью обратных связей, осуществляемых смежными узлами, с отрицательными весами передаются коэффициенты k_j загрузки этих узлов, а также (с положительными весами) оценки $r_j \leq 1$ ($j = 1, \dots, K$) состояния связи между данным и смежным узлами, которые окончательно влияют на выбор смежного узла для передачи пакета. (Значение k_j следует интерпретировать как достоверность высказывания о том, что буфер узла загружен полностью, а значение r_j — как достоверность высказывания о том, что качество связи между передающим и j -м узлами отличное.) Нейронная сеть фрагментарно распределена между компьютерами всех узлов так, чтобы отражать информацию, связанную только с конкретным узлом.

Фрагмент логической нейронной сети, размещенный на одном узле, представлен на рис. 3. Здесь ω_{ij} — предпочтительные веса смежных узлов при передаче пакета по адресу A_i , k_j — коэффициент загрузки буфера смежного узла, r_j — состояние связи между узлами.

Функция активации, по аналогии с (2) и (3) имеет вид

$$V_j = V_{Ai} \omega_{ij} - k_j + r_j,$$

если эта разность превышает порог h ;

0 в противном случае.

Здесь $V_{Ai} = 1$, если A_i ($i = 1, \dots, R$) — адрес назначения информационного пакета, порог h выбирается экспериментально так, чтобы предпочтение могло быть выбрано между не полностью загруженными узлами.

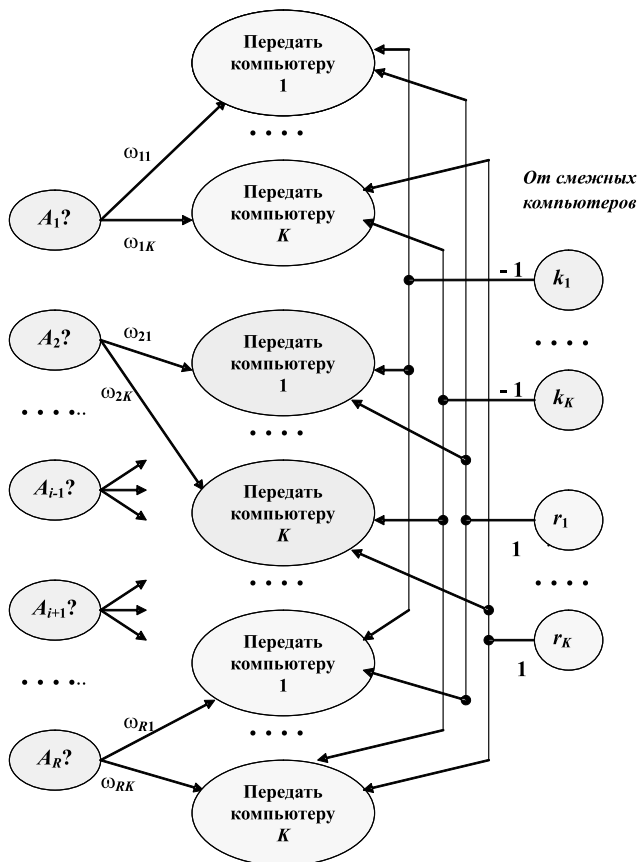


Рис. 3. Фрагмент логической нейронной сети, размещенный на одном узле, во взаимодействии со смежными узлами

Таким образом, в результате обратной связи максимального возбуждения может достичь совсем не тот смежный узел, которому первоначально было оказано предпочтение.

Общим критерием эффективности управления является максимизация пропускной способности сети. Частными критериями являются:

- минимум среднего времени выполнения запроса на передачу пакета в сети;
- минимум времени ожидания пользователем возможности выполнения своих запросов.

Рассмотренный подход легко распространяется на выбор оптимального множества маршрутов совместного следования объектов в транспортной сети.

Некоторые "незаконченные" системы принятия решений

Контроль космического пространства. Особую важность применение представленного метода ассоциативных вычислений обретает в системе контроля космического пространства. При трудностях политического характера непрерывное наблюдение всех космических объектов, особенно спутников, затруднительно. Как правило, спутник скрывается за горизонтом от средств обнаружения и сопровождения, а затем появляется вновь. Средства

обнаружения проводят селекцию и идентификацию на основе текущего состояния каталога, представляющего собой базу знаний о космическом пространстве. Новый объект должен быть внесен в каталог с измеренными параметрами.

Каталог строится в виде логической нейронной сети. Для каждого спутника за отдельными рецепторами закрепляются значения шести известных параметров орбит. (Если уже существует рецептор, закрепленный за необходимым значением, он используется повторно.) Связи от этих рецепторов с единичными весами идут на нейрон, "ответающий" за спутник, обладающий этими значениями параметров.

Тогда для вновь возникших из-за горизонта объектов методом ассоциативных вычислений с заданной точностью устанавливается совпадение с одним из известных спутников. Если спутник уже отображен в каталоге, т. е. возбуждение некоторого нейрона максимально превысило порог, продолжается его сопровождение. Если спутник новый, нейронная сеть дополняется на основе его характеристик, измеренных средствами обнаружения. Таким образом, логическая нейронная сеть, отображающая каталог, постоянно развивается.

Управление надежностью. Система управления надежностью сложной системы отображает решение трудноформализуемой задачи. В применении к железнодорожному транспорту [11], в условиях ограниченного финансирования, решение такой задачи влияет на порядок ремонта и модернизации объектов. Используется большое число контролируемых факторов: от прогнозируемой частоты отказов каждого транспортного средства до комплектации пожарных щитов, от срока эксплуатации железнодорожного перегона до наличия и обеспечения безопасности транспортного переезда и пассажирского перехода. Объединение показателей этих факторов в рамках единого метода ассоциативных вычислений возможно только при сведении их к единой системе измерения — к использованию достоверности высказываний о принадлежности данных: о наличии тех или иных объектов, о предполагаемых или прогнозируемых диапазонах надежности транспортных средств и т. д. Скорее всего, данная трудноформализуемая задача относится к задачам нахождения рейтинга каждого объекта в общей очереди всех объектов на обслуживание [12]. Рекомендуемые решения опираются не на строгие числовые оценки, которые могли бы быть результатом расчета функций численных переменных. Они указывают на предпочтительные стратегии развития и модернизации, на предпочтительный ряд возможных действий, распределение объема финансирования ремонта объектов и т. д.

Решение формируется в виде текстов, подобных тем, что выдаются медицинской информационно-справочной системой.

Данную систему принятия решений можно существенно развить, продлив ее логическую цепочку, введя второй логический уровень. На первом логическом уровне следует ограничиться нахождением текущего значения рисков, связанных с дальнейшей эксплуатацией объектов. Полученные данные совместно с другими данными, например, отображающими возможности финансирования, а также технологические и организационные возможности, следует использовать в следующей, однослойной логической нейронной сети — сети второго логического уровня. Эта сеть выдает окончательные конкретные рекомендации по тактике адресного проведения регламентных или ремонтных работ, по развитию инфраструктуры на основе новых требований, замене средств подвижного состава, прокладке путей и т. д. Примерный вид такой двухуровневой сети показан на рис. 4.

Реализация такой, всего лишь двухуровневой логической цепочки (длины, равной двум), должна послужить высокому уровню автоматизации и интеллектуализации системы управления, значительному подавлению человеческого фактора, повышению эффективности этой системы.

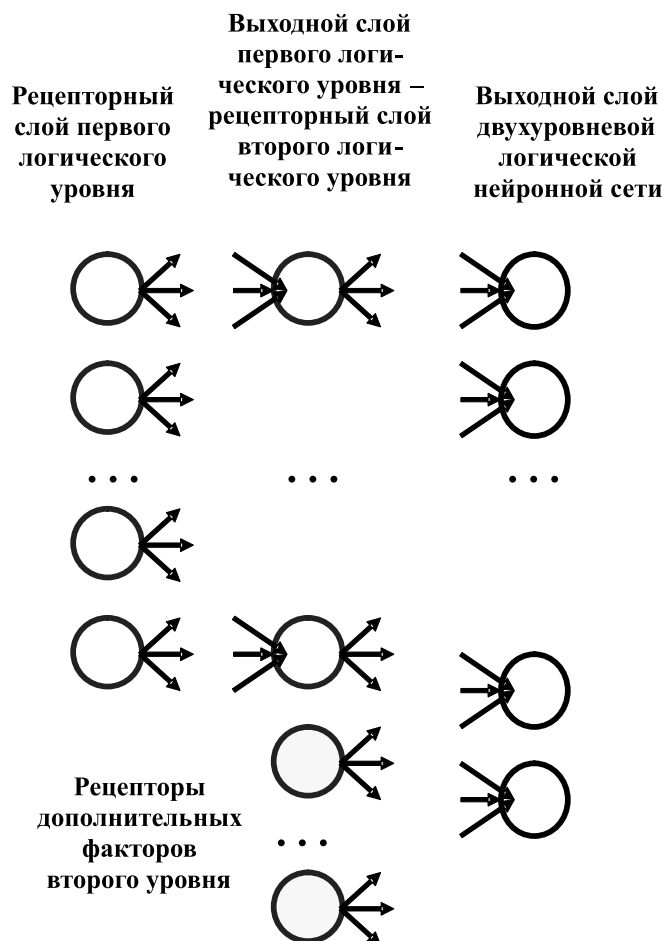


Рис. 4. Примерный вид логической нейронной сети, реализующей логическую цепочку с длиной, равной двум

Необходимо помнить, что обученные нейронные сети, реализующие систему принятия решений, строятся и сопровождаются на основе самого передового опыта управленцев и обобщения этого опыта экспертами на всю область возможного применения.

Применение матриц следования позволяет оперативно изменять и дополнять систему принятия решения новыми или уточненными связями и их весами на основе опыта эксплуатации. Так осуществляется обратная связь, позволяющая на основе эффективности принимаемых решений корректировать эти решения, т. е. неограниченно развивать базу знаний.

Самообучающиеся системы управления

Рассмотрим, по-видимому, единственный человеческий опыт построения самообучающейся системы управления на основе правил стрельбы наземной артиллерии времен Великой Отечественной войны.

Для расчета начальных установок при стрельбе батареи с закрытой огневой позиции стреляющий пользовался Таблицами стрельбы, составленными для каждого типа орудий при полигонных испытаниях. Такая таблица может служить прообразом базы знаний, реализуемой логической нейронной сетью.

Затем по цели или реперу следовала пристрелка одним орудием, при которой объект захватывался в широкую вилку, на границах которой наблюдались перелет и недолет. Эта вилка "половинилась" до достижения узкой вилки. Если объект был целью, на середине этой вилки давалась команда всем орудиям батареи на поражение.

После поражения цели или пристрелки репера следовала замечательная команда: "Стой! Записать...", при которой первый номер на щите орудия записывал *пристрелянные* установки по цели или реперу. Это проводилось для того, чтобы в последующем по близким (к пораженным) целям не выполнять трудоемкие расчеты и пристрелку в полном объеме, а использовать принцип ассоциативных вычислений для *переноса огня*, изменяя известные "близкие" установки. Так обеспечивалась возможность с большой вероятностью захватывать новую цель сразу в узкую вилку.

Таким образом, на деле реализовалась самообучающаяся система управления "стреляющий + батарея", позволяющая накапливать и оперативно использовать знания по всей местности, на всем доступном пространстве ведения боя — для переноса огня от пораженных целей или пристрелянных реперов.

Рассмотренный пример лежит в основе обобщения и построения развиваемой логической нейронной сети в составе самообучающейся системы управления и принятия решений. Подробно возможности и принципы построения самообучающихся систем управления изложены в работах [2, 4].

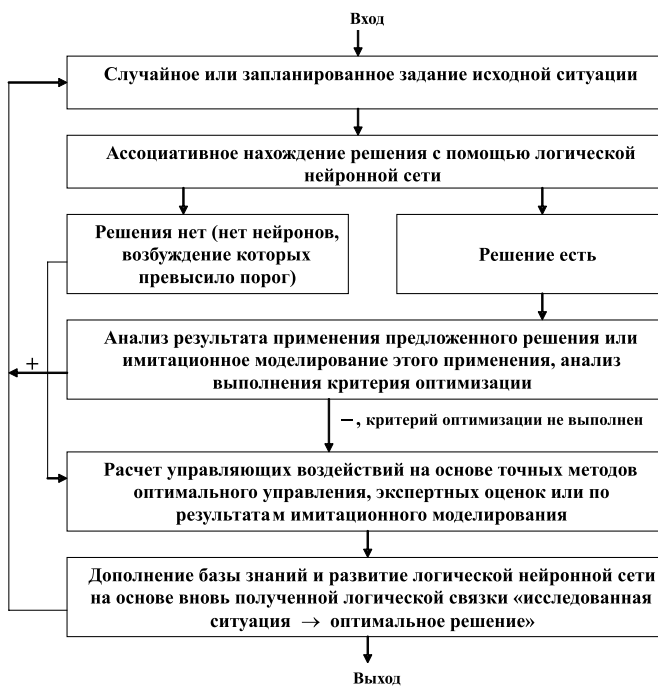


Рис. 5. Схема самообучения

Обобщим схему работы такой системы (рис. 5) в режиме обучения (подобно пристрелке реперов при занятии боевых позиций).

Работа многих сложных систем, особенно систем военного назначения, предполагает режим функционального контроля, когда система свободна от выполнения своих основных функций. В этом режиме и целесообразно развивать ее базу знаний — до полного насыщения.

Очевидно, что база знаний для рассматриваемого применения по форме не отличается от базы знаний информационно-справочной системы и большинства систем принятия решений. Поэтому логическая нейронная сеть имеет вид, представленный на рис. 1, 2, 5, и использует те же функции активации нейронов. Возможно включение обратных связей, как на рис. 3, а также формирование "длинных" логических цепочек дедуктивного вывода, как на рис. 4.

Заключение

Демонстрируется информационная технология построения системы принятия решений на логической нейронной сети для широкого круга применений. Становится очевидным, что применение

аппарата логических нейронных сетей расширяет возможности системы принятия решений, приводит к достижению их большей достоверности, к оперативной модификации сети в связи с модификацией структуры системы, позволяет оперативно учитывать новые знания, вводить и уточнять веса факторов, вводить новые факторы и их значения и т. д. Исключительно важной является возможность работы с нечеткими данными. Алгоритмы обработки логических нейронных сетей на основе примитивной пороговой функции активации обладают малой сложностью и не предъявляют высоких требований к производительности вычислительных средств. В то же время, подобно мозгу, они обладают широкими возможностями распараллеливания.

Принципы ассоциативных вычислений обеспечивают весьма простое задание исходных возбуждений рецепторов на основе приближенных оценок ситуации лицом, принимающим решение, и допускают оперативное исправление неправильных действий.

Список литературы

1. Барский А. Б. Нейронные сети: распознавание, управление, принятие решений. М.: Финансы и статистика. 2004. 175 с.
2. Барский А. Б. Логические нейронные сети: Учеб. пособие. М.: ИНТУИТ; БИНОМ. Лаборатория знаний, 2007. 352 с.
3. Барский А. Б. Математическая логика событий и логические нейронные сети // Информационные технологии. Приложение. 2007. № 7. 32 с.
4. Барский А. Б. Нейронные сети логического вывода. Курс лекций. Саарбрюккен: LAP LAMBERT Academic Publishing, 2011. 400 с.
5. Поспелов Д. А. Моделирование рассуждений. Опыт анализа мыслительных актов. М.: Радио и связь, 1989. 184 с.
6. Поспелов Д. А. Ситуационное управление. Теория и практика. М.: Наука. Гл. ред. физ.-мат. лит., 1986. 288 с.
7. Барский А. Б., Нгуен Ван Лой. Информационно-справочная система на логической нейронной сети "Многосерверная база данных с циркулирующими сегментами" // Информационные технологии. 2013. № 9. С. 57—61.
8. Барский А. Б., Дмитриев А. А., Барская О. А. Медицинские информационно-справочные системы на логических нейронных сетях // Информационные технологии. Приложение. 2010. № 1. 32 с.
9. Чжо Вин Латт. Логическая нейронная сеть в основе идентификации пользователя в системе "клиент — сервер" на базе трех характеристик "почерка" // Естественные и технические науки, 2010. № 3.
10. Барский А. Б., Саид Мохаммед М. Н. Выбор смещения при пошаговой маршрутизации в беспроводной сети // Мир Транспорта. 2013. № 2. С. 30—37.
11. Замышляев А. М. Прикладные информационные системы управления надежностью, безопасностью, рисками и ресурсами на железнодорожном транспорте. М.: "Журнал "Надежность". 2013. 143 с.
12. Барский А. Б. Рейтинговые системы на логических нейронных сетях // Информационные технологии. 2014. № 11. С. 66—72.

Learned and Self-Learning Recognition, Management and Decision-Making Systems on Base of the Logical Neural Networks

It is examined the logic neural networks application for associative "non-formula" hardly formulated tasks and simulation computing, in medical question-answered system, in user identification systems, in the step by step adaptive routing for wireless telecommunications and transport networks, in self-learning control systems by fuzzy data. In the base of logical neural networks lies the mathematical logic of events, combining the initial information from mathematical logic and probability theory. Excitation of receptors is based on the reliability of statements about data ownership, which provides a single system of measurement of baseline data and dramatically extends the scope of application of neural networks to construct almost any decision-making system that combines the data of any nature, types and forms of representation. Excitation of neurons in the output layer indicates a decision to be made in the form of value, the text of the control signal, and so on. Networks are reduced to a single layer. In these learning situations directly linked to corresponding solutions. Logic neural networks are differing by low requirements for the qualification developer, simple calculations, high speed and the possibility of parallelization. Guidelines for a better incarnation of some of the known decision-making systems are provides.

Keywords: logic neural network, the activation function, the matrix sequence, associative computing, information and referral system, self-learning management system

References

1. Barskij A. B. *Nejronnye seti: raspoznavanie, upravlenie, prinyatie reshenij*. M.: Finansy i statistika, 2004. 175 p.
2. Barskij A. B. *Logicheskie nejronnye seti*. M.: INTUIT; BINOM. Laboratoriya znaniy, 2007. 352 p.
3. Barskij A. B. Matematicheskaja logika sobytij i logicheskie nejronnye seti. *Informacionnye tehnologii. Prilozhenie*, 2007, no. 7. 32 p.
4. Barskij A. B. *Nejronnye seti logicheskogo vyvoda*. Kurs lekczij. Saarbrücken: LAP LAMBERT Academic Publishing, 2011. 400 p.
5. Pospelov D. A. *Modelirovanie racyzhdenij. Opyt analiza myslitelnyh aktov*. M.: Radio i svyaz', 1989. 184 p.
6. Pospelov D. A. *Situacionnoe upravlenie. Teoriya i praktika*. M.: Nauka. Gl. red. Fiz.-mat. Lit, 1986. 288 p.
7. Barskij A. B., Nguen Van Loj. Informacionno-spravochnaja sistema na logicheskoj nejronnoj seti "Mnogoservernaya baza dannyh s czirkulirujushhimi segmentami". *Informacionnye tehnologii*. 2013, no. 9. P. 57–61.
8. Barskij A. B., Dmitriev A. A., Barskaya O. A. Meditsinskie informacionno-spravochnye sistemy na logicheskikh nejronnyh setyah. *Informacionnye tehnologii. Prilozhenie*, 2010, no. 1. 32 p.
9. Chzho Vin Latt. Logicheskaya nejronnaya set' v osnove identifikacii pol'zovatelya v sisteme "klient — server" na baze tryoh harakteristik "pocherka". *Estestvennye i tehicheskie nauki*, 2010, no. 3.
10. Barskij A. B., Said Mohammed M. N. Vybora smeshheniya pri poshagovoj marshrutizacii v besprovodnoj seti. *Mir Transporta*, 2013, no. 2, pp. 30–37.
11. Zamyshlyayev A. M. *Prikladnye informacionnye sistemy upravleniya nadyozhnost'yu, bezopasnost'yu, riskami i resursami na zhelezno-dorozhnom transporte*. M.: "Zhurnal "Nadyozhnost'", 2013. 143 p.
12. Barskij A. B. Rejtingovye sistemy na logicheskikh nejronnyh setyah. *Informacionnye tehnologii*. 2014, no. 11, pp. 66–72.

УДК 004.942

Л. С. Куравский, д-р техн. наук, проф., декан, e-mail: l.s.kuravsky@gmail.com,
П. А. Мармалюк, канд. техн. наук, доц., Г. А. Юрьев, канд. физ.-мат. наук, доц., П. Н. Думин, аспирант,
Московский городской психолого-педагогический университет,
факультет информационных технологий

Методы численной идентификации марковских моделей и их сравнительный анализ¹

Представлены численные методы нулевого и первого порядка, предназначенные для идентификации марковских процессов с дискретными состояниями и непрерывным временем по результатам наблюдений, а также технология вычислительного эксперимента, обеспечивающего сравнение характеристик эффективности рассматриваемых подходов. Анализ результатов проведенных экспериментов показал, что разработанные методы идентификации имеют преимущества перед классическим градиентным методом первого порядка. Рассмотренные алгоритмы можно применять для настройки параметров обучаемых структур, включая нейронные сети.

Ключевые слова: марковские модели, идентификация моделей, многомерная нелинейная оптимизация, обучаемые структуры

¹ Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 14-06-00191) и Российского гуманитарного научного фонда (проект № 14-06-12012).

1. Постановка задачи идентификации

Поведение многих технических, диагностических и других систем представляется параметрическими математическими моделями в форме систем обыкновенных дифференциальных уравнений. В частности, такая форма представления применялась для моделей усталостного разрушения и тестирования, построенных на базе *марковских случайных процессов с дискретными состояниями и непрерывным временем* [1–6, 11]. Важной особенностью практического использования марковских моделей в указанных задачах является то, что их параметры идентифицируются по результатам наблюдений, представленных наборами значений исследуемых функций в контрольных точках.

Марковские модели для описания динамики переходов между состояниями представляются ориентированными графами, в которых вершины соответствуют состояниям, а дуги — переходам, для которых выполняются свойства пуассоновских потоков событий. В этих потоках число событий N , попадающих в любой временной интервал длины τ , начинающийся в момент t , распределено согласно закону Пуассона:

$$P_{t,\tau}(N = r) = \frac{a(t, \tau)^r}{r!} e^{-a(t, \tau)},$$

где $P_{t,\tau}(X = r)$ — вероятность появления r событий в течение рассматриваемого интервала; $a(t, \tau)$ — среднее число событий, попадающих в интервал длины τ , начинающийся в момент времени t . Рассматриваются только стационарные потоки, в которых $a(t, \tau) = \eta\tau$, а $\eta = \text{const}$ есть интенсивность стационарного потока. Упомянутые выше предположения о свойствах потоков событий обычны для прикладных задач [7], так как эти потоки (или потоки, близкие к ним по свойствам) часто встречаются на практике благодаря предельным теоремам для потоков событий.

Полагается, что:

- для указанных процессов с $n + 1$ дискретными состояниями и непрерывным временем заданы начальные распределения вероятностей и наблюдаемые частоты пребывания в состояниях процессов $\{F_{id}\}_{i=0, \dots, n}$ в моменты времени $\{t_d\}_{d=0, \dots, D-1}$, где D — количество моментов времени, в которые фиксировались частоты F_{id} ; $0 \leq t_d \leq T$, T — конечный момент времени;

- интенсивности переходов между состояниями полностью или частично являются неизвестными (свободными) параметрами.

Динамика изменения вероятностей пребывания в состояниях процесса определяется системой обыкновенных дифференциальных уравнений Колмогорова в матричной форме:

$$\frac{dp(t)}{dt} = \mathbf{M}(\lambda)\mathbf{p},$$

где $0 \leq t \leq T$, $\mathbf{p}(t) = (p_0(t), \dots, p_n(t))^T$ — вероятности пребывания в состояниях процесса; $\lambda = (\lambda_0, \dots, \lambda_m)^T$ — упорядоченный набор интенсивностей переходов между состояниями, $n > m$; \mathbf{M} — матрица интенсивностей переходов между состояниями порядка $n + 1$.

Для указанной системы уравнений ставится задача идентификации набора параметров λ . Значения параметров из этого набора определяются путем сравнения наблюдаемых и прогнозируемых гистограмм, описывающих распределения частот пребывания в состояниях модели, а именно: вычисляются значения, обеспечивающие наилучшее соответствие наблюдаемых и ожидаемых частот попадания в определенное состояние системы в заданные моменты времени. При этом определяется набор интенсивностей λ , обеспечивающий наименьшее значение статистики Пирсона:

$$X^2(\lambda) = \sum_{d=0}^{D-1} \sum_{i=0}^n \frac{(p_i(\lambda t_d)N - F_{id})^2}{p_i(\lambda t_d)N},$$

где $N = \sum_{i=0}^n F_{id}$. Эта статистика используется как

мера соответствия модели наблюдениям.

Доказано, что при выполнении ряда общих условий значения указанной статистики X^2 , получаемые при подстановке истинных решений, асимптотически описываются распределением χ^2 с $n - m$ степенями свободы, где m — число определяемых параметров, причем вычисленные значения свободных параметров при увеличении объема выборки сходятся по вероятности к искомому решению [8]. Это позволяет использовать статистику Пирсона для проверки гипотезы о том, что полученный прогноз согласуется с результатами наблюдений.

Таким образом, постановка задачи предполагает решение обратной задачи, в которой коэффициенты дифференциальных уравнений определяются по фрагментам решения, заданным наблюдениями. Представленную выше постановку можно формально рассматривать как прямую задачу с усложненным вычислением критерия X^2 посредством решения задачи Коши для системы обыкновенных дифференциальных уравнений, которая решается стандартными методами оптимизации.

Для ее решения, в частности, допустимы алгоритмы локальной оптимизации, включая градиентные методы, и стохастические алгоритмы. Однако опыт их практического применения позволяет говорить о неприемлемо большом времени вычислений в случае задач высокой размерности (при больших значениях параметра m), а также о проблемах, обусловленных возможной неединственностью получаемого решения.

Для решения этих проблем предложены представленные далее в подразделах 2.2 и 2.3 численные методы [10], предназначенные для идентификации набора параметров λ , определяющего матрицу \mathbf{M} .

Особенностью этих методов является использование начальных приближений, вычисляемых по результатам наблюдений, а также оценок чувствительности критерия оптимизации к малым изменениям идентифицируемых параметров. Новым элементом второго из предложенных подходов является улучшение временных характеристик применяемого комбинаторного алгоритма подбора значений идентифицируемых переменных за счет замены на каждой итерации вычислений перебора различных сочетаний этих значений в окрестности их текущих оценок перебором значений только тех переменных, к изменению которых наиболее чувствителен критерий оптимизации. Для оценки этой чувствительности используют разностные аппроксимации соответствующих частных производных. Метод, рассмотренный в подразделе 2.2, относится к методам нулевого порядка, а его модифицированный вариант из подраздела 2.3 — к методам первого порядка. Характеристики их эффективности сопоставляют с аналогичными показателями классического подхода, представленного градиентным методом первого порядка из подраздела 2.1.

Рассмотренные алгоритмы можно эффективно применять для настройки параметров обучаемых структур, включая нейронные сети.

2. Сравнимые методы оптимизации

2.1. Градиентный метод: алгоритм вычислений

1. Используя имеющиеся результаты наблюдений, вычислить в качестве начальных оценок параметров λ_i ($i = 0, \dots, m$), обозначаемых как λ_i^0 ($i = 0, \dots, m$), выборочные средние значения числа переходов между парами соответствующих состояний марковского процесса в единицу времени (или, что иногда удобнее, величины, обратные выборочным средним продолжительностям времени между двумя смежными переходами для этих же пар состояний). Полученные оценки рассматривать как начальные приближения к идентифицируемым параметрам.

2. Задать счетчик итераций $j = 0$.

3. Задать начальный шаг градиентного метода h .

4. В точке текущего приближения λ^j вычислить оценку градиента $\nabla X^2(\lambda)$ с использованием центрально-разностной аппроксимации, при этом значения $\mathbf{p}(\lambda, t_{ij})$ определить путем численного решения

задачи Коши для системы уравнений $\frac{d\mathbf{p}}{dt} = \mathbf{M}(\lambda)\mathbf{p}$

с заданными начальными условиями.

5. Выполнить коррекцию текущего приближения:

$$\lambda^{j+1} = \lambda^j - h\nabla X^2(\lambda^j).$$

6. Если $X^2(\lambda^{j+1}) < X^2(\lambda^j)$, $\|\lambda^j - \lambda^{j+1}\| > \varepsilon$ и не исчерпан лимит итераций, то положить $j = j + 1$ и перейти к шагу 3, иначе перейти к шагу 6.

7. Если $X^2(\lambda^{j+1}) \geq X^2(\lambda^j)$, $\|\lambda^j - \lambda^{j+1}\| > \varepsilon$ и не исчерпан лимит итераций, то положить $h = \frac{h}{2}$ и перейти к шагу 4, иначе завершить вычисления.

2.2. Метод полного перебора: алгоритм вычислений

1. Используя имеющиеся результаты наблюдений, вычислить в качестве начальных оценок параметров λ_i ($i = 0, \dots, m$), обозначаемых как λ_i^0 ($i = 0, \dots, m$), выборочные средние значения числа переходов между парами соответствующих состояний марковского процесса в единицу времени (или, что иногда удобнее, величины, обратные выборочным средним продолжительностям времени между двумя смежными переходами для этих же пар состояний). Полученные оценки рассматривать как начальные приближения к идентифицируемым параметрам.

2. Положить $j = 1$, $k = 1$, $\beta_1 = 1 + \gamma_1$, где γ_1 — параметр алгоритма.

3. Для каждой из текущих оценок идентифицируемых параметров λ_i^j ($i = 0, \dots, m$) на j -й итерации алгоритма вычислить верхнюю λ_i^{j+} и нижнюю λ_i^{j-} границу сдвига оценки по формулам $\lambda_i^{j+} = \beta_j \lambda_i^j$, $\lambda_i^{j-} = \beta_j^{-1} \lambda_i^j$.

4. Выполнив полный перебор всех вариантов текущих оценок набора параметров $\lambda_j = (\lambda_0^j, \dots, \lambda_m^j)^T$, в которых каждый из его компонентов λ_i^j ($i = 0, \dots, m$) принимает только три возможных значения из множества $\{\lambda_i^{j-1,-}, \lambda_i^{j-1}, \lambda_i^{j-1,+}\}$, выбрать из указанных вариантов одну из оценок $\lambda_{j,*} = (\lambda_{0,*}^j, \dots, \lambda_{m,*}^j)^T$, обеспечивающих минимальное значение критерия X^2 , равное $X_{j,\min}^2$.

5. Если для всех компонентов выбранного набора $\lambda_{j,*}$ выполняется равенство $\lambda_{i,*}^j = \lambda_i^{j-1}$ ($i = 0, \dots, m$), то положить $k = k + 1$, $\gamma_k = \gamma_{k-1}/2$.

6. Положить $j = j + 1$, $\beta_j = 1 + \gamma_k$, $\lambda_j = \lambda_{j-1,*}$.

7. Если $X_{j-1,\min}^2 > X_*^2$, где X_*^2 — параметр алгоритма, то перейти к шагу 3, иначе завершить вычисления.

Временная сложность одной итерации рассмотренного алгоритма есть $O(3^{m+1})$, что приводит к неприемлемо длительным вычислениям при больших значениях m . Для того чтобы временная сложность стала приемлемой для практических вычислений, перебор вариантов текущих оценок набора

параметров $\lambda_j = (\lambda_0^j, \dots, \lambda_m^j)^T$ на шаге 4 приведенного выше алгоритма следует осуществлять не для всех параметров λ_i^j ($i = 0, \dots, m$), а только для их подмножества, состоящего из $l < m + 1$ элементов, к изменению которых наиболее чувствителен критерий X^2 .

2.3. Метод перебора значимых параметров: алгоритм вычислений

1. Используя имеющиеся результаты наблюдений, вычислить в качестве начальных оценок параметров λ_i ($i = 0, \dots, m$), обозначаемых как λ_i^0 ($i = 0, \dots, m$), выборочные средние значения числа переходов между парами соответствующих состояний марковского процесса в единицу времени (или, что иногда удобней, величины, обратные выборочным средним продолжительностям времени между двумя смежными переходами для этих же пар состояний). Полученные оценки рассматривать как начальные приближения к идентифицируемым параметрам.

2. Положить $j = 1, k = 1, \beta_1 = 1 + \gamma_1$, где γ_1 — параметр алгоритма.

3. Оценить чувствительность критерия X^2 в малой окрестности каждой из текущих оценок идентифицируемых параметров λ_i^j ($i = 0, \dots, m$) на j -й итерации алгоритма, используя для этого разностные аппроксимации абсолютных значений частных производных $\delta_i(\varepsilon) = |X^2(\lambda_0^j, \dots, \lambda_i^j + \varepsilon, \dots, \lambda_m^j) - X^2(\lambda_0^j, \dots, \lambda_i^j, \dots, \lambda_m^j)|/\varepsilon$, где $i = 0, \dots, m$; ε — параметр алгоритма.

4. Выбрать l текущих оценок идентифицируемых параметров $\lambda_{i_q}^j$ (где $i_q \in \{0, \dots, m\}; q = 1, \dots, l$; l — параметр алгоритма); имеющих наибольшие значения оценок чувствительности $\delta_i(\varepsilon)$.

5. Для каждой из выбранных на шаге 4 текущих оценок идентифицируемых параметров $\lambda_{i_q}^j$ на j -й итерации алгоритма, где $i_q \in \{0, \dots, m\}; q = 1, \dots, l$, вычислить верхнюю $\lambda_{i_q}^{j+}$ и нижнюю $\lambda_{i_q}^{j-}$ границу сдвига оценки по формулам $\lambda_{i_q}^{j+} = \beta_j \lambda_{i_q}^j, \lambda_{i_q}^{j-} = \beta_j^{-1} \lambda_{i_q}^j$.

6. Выполнив полный перебор всех вариантов текущих оценок набора параметров $\lambda_j = (\lambda_0^j, \dots, \lambda_m^j)^T$, в которых каждый из его компонентов $\lambda_{i_q}^j$, где $i_q \in \{0, \dots, m\}; q = 1, \dots, l$, принимает только три возможных значения из множества $\{\lambda_{i_q}^{j-1,-}, \lambda_{i_q}^{j-1}, \lambda_{i_q}^{j-1,+}\}$, а оставшиеся компоненты принимают значение

своей текущей оценки на j -й итерации алгоритма, выбрать из указанных вариантов одну из оценок $\lambda_{j,*} = (\lambda_{0,*}^j, \dots, \lambda_{m,*}^j)^T$, обеспечивающих минимальное значение критерия X^2 , равное $X_{j,\min}^2$.

7. Если для всех компонентов выбранного набора $\lambda_{j,*}$ выполняется равенство $\lambda_{i,*}^j = \lambda_{i,*}^{j-1}$ ($i = 0, \dots, m$), то положить $k = k + 1, \gamma_k = \gamma_{k-1}/2$.

8. Положить $j = j + 1, \beta_j = 1 + \gamma_k, \lambda_j = \lambda_{j-1,*}$.

9. Если $X_{j-1,\min}^2 > X_*^2$, где X_*^2 — параметр алгоритма, то перейти к шагу 3, иначе завершить вычисления.

Для решаемых на практике типовых задач общее время вычислений зависит от параметра l монотонно, достигая наименьшего значения при $l = 1$. Временная сложность одной итерации модифицированного алгоритма при $l = 1$ есть $O(m)$, что свидетельствует об асимптотически линейной зависимости времени вычислений, необходимого для выполнения одной итерации, от числа идентифицируемых параметров.

3. Генерация пула данных для вычислительного эксперимента

Для оценки эффективности предложенных алгоритмических решений был создан пул тестовых структур, близких по своим параметрам и структуре к моделям, используемым при решении реальных прикладных задач. В качестве базовой рассматривалась марковская модель, организованная по схеме "гибели и размножения". В процессе моделирования управляемыми параметрами были число состояний такой модели и число независимых параметров, через которые выражались интенсивности переходов между состояниями.

Самым простым случаем является ситуация, при которой все интенсивности λ равны между собой, а самым сложным — случай, при котором каждый из переходов имеет собственную интенсивность. В процессе генерации моделей из заданного числового интервала случайным образом выбирались значения переменных, соответствующие интенсивностям переходов между состояниями.

Пример типовой модели представлен на рис. 1 в виде ориентированного графа.

Структура связей подобного графа может быть описана матрицей смежности, элементы которой содержат значения интенсивностей переходов (рис. 2).

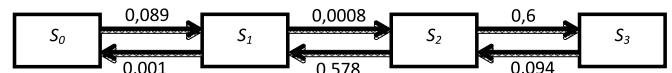


Рис. 1. Организованная по схеме "гибели и размножения" марковская модель из четырех состояний. Интенсивности переходов определяются шестью независимыми параметрами

	i	S_0	S_1	S_2	S_3
j	S_0		0,089		
	S_1	0,001		0,0008	
	S_2		0,578		0,6
	S_3			0,094	

Рис. 2. Пример матрицы смежности, отражающей структуру связей и интенсивности переходов, для модели с четырьмя состояниями и шестью независимыми параметрами

Матрица, представленная на рис. 2, позволяет сформировать следующую систему обыкновенных дифференциальных уравнений Колмогорова:

$$\frac{dP_{S_0}(t)}{dt} = P_{S_1}(t) \cdot 0,001 - P_{S_0}(t) \cdot 0,089;$$

$$\begin{aligned} \frac{dP_{S_1}(t)}{dt} = & P_{S_0}(t) \cdot 0,089 + P_{S_2}(t) \cdot 0,578 - \\ & - P_{S_1}(t) \cdot (0,001 + 0,0008); \end{aligned}$$

$$\begin{aligned} \frac{dP_{S_2}(t)}{dt} = & P_{S_2}(t) \cdot 0,0008 + P_{S_3}(t) \cdot 0,094 - \\ & - P_{S_2}(t) \cdot (0,578 + 0,6); \end{aligned}$$

$$\frac{dP_{S_3}(t)}{dt} = P_{S_2}(t) \cdot 0,6 - P_{S_3}(t) \cdot 0,094.$$

Для всех созданных моделей в начальный момент времени t_0 вероятность пребывания в состоянии S_0 была равна единице, а во всех прочих — нулю. Для каждой возможной комбинации "Число состояний/число параметров" (до 10 состояний) было сгенерировано по 20 примеров (рис. 3).

Для систем уравнений Колмогорова, соответствующих сгенерированным моделям, на временном интервале $[0; 1000]$ численно решалась задача

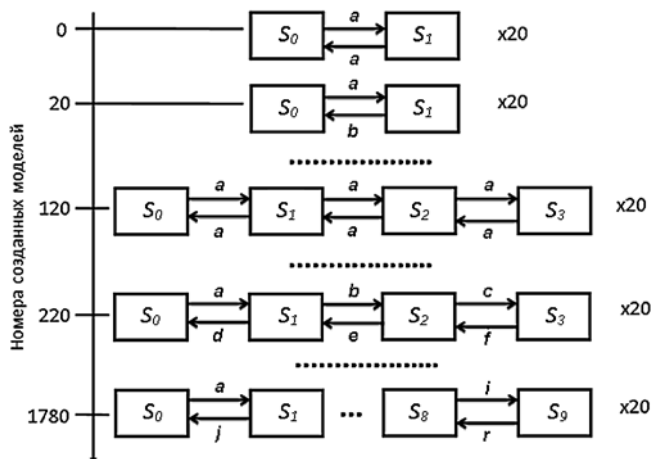


Рис. 3. Структура сформированного пула моделей для идентификации

Коши [9]. Вычисленные значения вероятностей $\mathbf{P}(t) = (p_{S_0}(t), \dots, p_{S_n}(t))^T$ для контрольных точек $\{t_d\}_{d=0, \dots, D-1}$ сохранялись в файле модели.

Следует отметить, что в ряде комбинаций подобный подход к генерации моделей приводит к нарушению указанного в разд. 1 условия $n > m$. Использование статистики χ^2 в качестве критерия качества модели при этом становится невозможным. В таких случаях рекомендуется использовать рассмотренный в работе [1] критерий, основанный на оценке значимости компонентов уравнения регрессии, связывающего наблюдаемые и прогнозируемые частоты.

Чтобы избежать ряда вычислительных проблем, связанных с представлением малых вероятностей, вместо χ^2 в качестве минимизируемого критерия использовали статистику $Q = \sum_{d=0}^{D-1} \sum_{i=0}^n (p_i(t_d, \lambda) - \tilde{p}_i(t_d))^2$,

где $\tilde{p}_i(t_d)$ — вероятность пребывания в i -м состоянии модели в момент времени t_d , полученная при интегрировании уравнений модели на этапе ее генерации.

Для вычислительного эксперимента было сгенерировано 1800 моделей, каждая из которых представлялась XML-файлом, содержащим: матрицу смежности; значения независимых параметров, определяющих интенсивности переходов; параметры, используемые при численном интегрировании уравнений модели; имена состояний; значения вероятностей $\mathbf{P}(t) = (p_{S_0}(t), \dots, p_{S_n}(t))^T$ в контрольных точках $\{t_d\}_{d=0, \dots, D-1}$.

Таблицы 1 и 2 содержат число сгенерированных моделей для каждого набора параметров и каждого числа состояний.

Таблица 1		Таблица 2	
Число моделей, сгенерированных для каждого числа параметров		Число моделей, сгенерированных для каждого числа состояний	
Число параметров	Число моделей	Число состояний	Число моделей
1	180	2	40
2	180	3	80
3	160	4	120
4	160	5	160
5	140	6	200
6	140	7	240
7	120	8	280
8	120	9	320
9	100	10	360
10	100		
11	80		
12	80		
13	60		
14	60		
15	40		
16	40		
17	20		
18	20		

4. Параметры вычислительной системы

Программное обеспечение для вычислительного эксперимента реализовано на языке *G* в среде графического программирования *National Instruments LabVIEW 2010*. Характеристики системы, работавшей под управлением ОС *Microsoft Windows 7 Professional (SP1)*, следующие:

- материнская плата ASRock Z77 Extreme4-M Socket LGA 1155 (1066—2800 МГц);
- четырехъядерный процессор Intel Core i5-3330 с частотой 3000 МГц;
- два модуля оперативной памяти DDR3 Corsair XMS3 объемом 4 Гбайт с частотой 1333 МГц.

5. Параметры процедур идентификации

Поскольку при генерации пула данных для вычислительного эксперимента отсутствует информация, необходимая для определения начальных оценок интенсивностей переходов, оценки параметров λ_i^0 на первом шаге представленных выше алгоритмов устанавливались равными заданным константам. Значение параметра γ_1 , определяющего сдвиг от текущих оценок идентифицируемых величин, устанавливалось равным 0,9. Число контрольных моментов времени D было равно 100. По достижении значения Q , удовлетворяющего неравенству

$$\frac{Q}{(n+1)D} \leq 0,001, \text{ поиск останавливался, а соответствующая модель считалась идентифицированной.}$$

Время вычислений, затрачиваемое на каждую модель, ограничивалось 200 с. По его истечении поиск останавливался, а соответствующая модель считалась неидентифицированной. В случаях методов полного перебора и перебора двух значимых параметров идентификацию проводили только для моделей, имеющих не более шести параметров.

6. Результаты вычислительного эксперимента

После идентификации каждой модели характеристики проведенного процесса вычислений сохранялись в виде *XML*-файла. Сформированный таким образом набор данных включал название метода идентификации, число состояний, число параметров модели, временные отметки для каждой итерации и другую необходимую информацию.

Созданные файлы использовали для формирования выборки результатов вычислительных экспериментов со следующими характеристиками: число состояний *States* и параметров *Params* модели, идентификационный номер модели *Trial*, евклидово расстояние *Distance* начальных оценок параметров до их значений, заложенных при генерации модели, число итераций *Iterations*, выполненных до достижения одного из критериев останова, результирующее значение *ResFit* минимизируемого критерия,



Рис. 4. Зависимости доли неидентифицированных моделей от числа параметров моделей при разных методах идентификации

целевое значение *GoalFit* минимизируемого критерия, булев флаг *MissFitFlag*, определяющий успешность идентификации модели (0 — модель успешно идентифицирована, 1 — модель не идентифицирована), номинальный фактор *Alg* с пятью значениями (0 — градиентный метод, 1 — полный перебор, 2 — перебор двух значимых параметров, 3 — перебор трёх значимых параметров, 4 — перебор четырёх значимых параметров). Полученные данные анализировали в программной среде для статистических вычислений *R* [12].

Зависимости доли моделей, которые не были идентифицированы вследствие превышения заданного лимита времени, от числа параметров моделей при разных методах идентификации представлены на рис. 4. Эти результаты свидетельствуют о том, что разработанные методы идентификации, за исключением метода полного перебора при более чем пяти параметрах, обеспечивают существенное ускорение вычислений по сравнению с рассмотренным классическим градиентным методом.

На рис. 5 представлены графики типа "ящик с усами", отражающие зависимости статистических характеристик времени вычислений (размаха, межквартильного интервала и медианы) от числа состояний и используемого метода идентификации (слева) и от числа параметров модели и используемого метода идентификации (справа). Вдоль горизонтальной оси на этом и последующем рисунках указаны составные показатели, первый компонент которых есть число состояний или параметров, а второй — индекс метода (1 — полный перебор, 2 — перебор двух значимых параметров). Приведенные графики иллюстрируют показательный характер зависимости времени вычислений от числа параметров модели для метода идентификации, основанного на полном переборе комбинаций оценок,

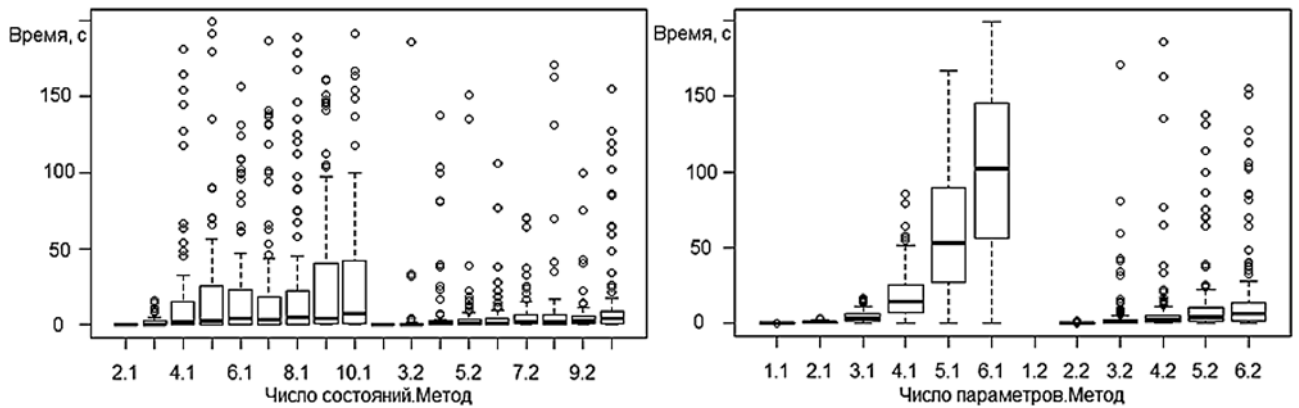


Рис. 5. Зависимости статистических характеристик времени вычислений (размаха, межквартильного интервала и медианы) от числа состояний и параметров и используемого метода идентификации (1 — полный перебор; 2 — перебор двух значимых параметров)

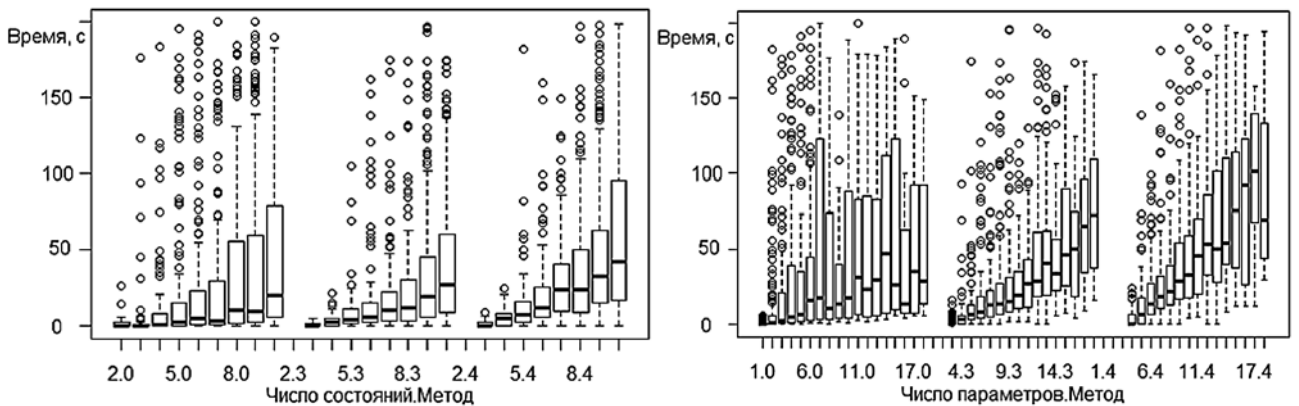


Рис. 6. Зависимости статистических характеристик времени вычислений (размаха, межквартильного интервала и медианы) от числа состояний и параметров и используемого метода идентификации (0 — градиентный метод, 3 — метод перебора трех значимых параметров, 4 — метод перебора четырех значимых параметров)

и близкий к линейному характер такой зависимости в случае перебора двух значимых параметров.

Те же самые статистические характеристики для градиентного метода и методов перебора трех и четырех значимых параметров, обозначенных, соответственно, индексами 0, 3 и 4, представлены на рис. 6. Динамика роста медианы времени вычислений на приведенных графиках позволяет говорить о том, что градиентный метод наименее подвержен влиянию числа идентифицируемых параметров. Однако зависимости, представленные на рис. 4, свидетельствуют, что этот метод в несколько раз чаще, чем другие подходы, не обеспечивает нахождение решения (если число параметров больше восьми, решение не определяется примерно в 34...55 % случаев). Это существенно ограничивает практическую применимость градиентного метода и, по-видимому, объясняется известной чувствительностью этого способа решения к выбору начальных приближений.

Для удобства сопоставления рассмотренных методов идентификации на рис. 7 отдельно приведены зависимости среднего времени вычислений от числа параметров моделей. Представленные результаты свидетельствуют о том, что разработанные методы

идентификации имеют преимущества перед классическим градиентным методом при числе идентифицируемых параметров, не превышающем 15. Если число указанных параметров превышает 15, то градиентным методом решение получается не-

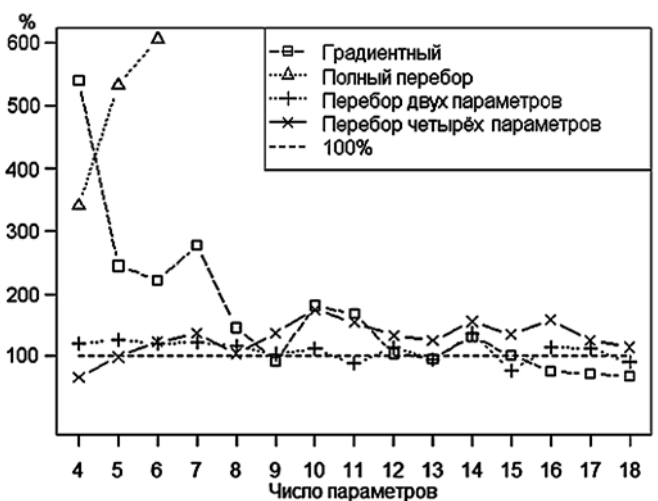


Рис. 7. Зависимости доли среднего времени вычислений относительно среднего времени, затраченного на вычисления методом перебора трех параметров, от числа параметров моделей для различных методов идентификации

Таблица 3

Статистические оценки значимости влияний различных характеристик на время вычислений, полученные путем анализа заданной линейной регрессионной зависимости

Статистика $R^2 = 0,26$; F -статистика = 309,3 при 7 и 6026 степенях свободы, $p < 2,2 \cdot 10^{-16}$					
Параметр регрессионной модели	Оценка коэффициента	Стандартная ошибка	t -статистика	p -значение	Статистическая значимость
b_0	-5,94	1,88	-3,15	0,00165	Есть
$States$	1,65	0,24	6,86	$7,95 \cdot 10^{-12}$	Есть
$Params$	3,95	0,14	27,27	$< 2,0 \cdot 10^{-16}$	Есть
$Distance$	14,35	1,34	10,7	$< 2,0 \cdot 10^{-16}$	Есть
$Alg1$	1,52	1,65	0,91	0,36	Нет
$Alg2$	-17,55	1,45	-12,1	$< 2,0 \cdot 10^{-16}$	Есть
$Alg3$	-21,43	1,5	-14,29	$< 2,0 \cdot 10^{-16}$	Есть
$Alg4$	-12,71	1,56	-8,16	$4,21 \cdot 10^{-16}$	Есть

сколько быстрее, однако примерно в 35—55 % случаев его сходимость не гарантирована (см. рис. 4), и решение за разумное время не удастся получить.

В табл. 3 представлены статистические оценки значимости влияний различных характеристик на время вычислений, полученные для следующей линейной регрессионной зависимости:

$$Time = b_0 + b_1 States + b_2 Params + b_3 Distance + b_4 Alg1 + b_5 Alg2 + b_6 Alg3 + b_7 Alg4,$$

где $Time$ — время вычислений; $States$ — число состояний модели; $Params$ — число параметров модели; $Distance$ — евклидово расстояние от начальных оценок параметров до их оптимальных значений; $Alg1, Alg2, Alg3, Alg4$ — бинарные показатели, принимающие значение 1 для результатов, полученных, соответственно, полным перебором, перебором двух значимых параметров, перебором трех значимых параметров и перебором четырех значимых параметров, и принимающие значение 0 — в противном случае.

Данные, представленные в табл. 3, свидетельствуют о том, что рассмотренная регрессионная модель в целом, и все ее компоненты статистически значимы. Анализ коэффициентов, соответствующих бинарным показателям, представляющим метод идентификации, позволяет говорить о существенном ускорении вычислений при использовании методов перебора значимых параметров по сравнению с классическим градиентным методом.

В целом рассмотренные выше результаты вычислительных экспериментов позволяют говорить о заметных преимуществах разработанных методов идентификации перед классическим градиентным методом.

7. Основные результаты и выводы

1. Разработаны численные методы нулевого и первого порядка, предназначенные для идентификации марковских процессов с дискретными состояниями и непрерывным временем по результатам наблюдений.

2. Разработана технология вычислительного эксперимента в целях сравнения характеристик эффективности рассматриваемых подходов к численному решению задачи идентификации. Ее программная реализация включает средства для генерации пула марковских моделей, исследования их поведения и анализа получаемых результатов.

3. Анализ результатов проведенных вычислительных экспериментов показал, что разработанные методы идентификации имеют преимущества перед классическим градиентным методом первого порядка.

Список литературы

1. Куравский Л. С., Марголис А. А., Юрьев Г. А., Мармалюк П. А. Концепция системы поддержки принятия решений для психологического тестирования // Психологическая наука и образование. 2012. № 1. С. 56—65.
2. Куравский Л. С., Мармалюк П. А., Алхимов В. И., Юрьев Г. А. Математические основы нового подхода к построению процедур тестирования // Экспериментальная психология. 2012. Том 5, № 4. С. 75—98.
3. Куравский Л. С., Марголис А. А., Мармалюк П. А., Юрьев Г. А., Думин П. Н. Обучаемые марковские модели в задачах оптимизации порядка предъявления психологических тестов // Нейрокомпьютеры: разработка и применение. 2013. № 4. С. 28—38.
4. Куравский Л. С., Мармалюк П. А., Алхимов В. И., Юрьев Г. А. Новый подход к построению интеллектуальных и компетентностных тестов // Моделирование и анализ данных. 2013. № 1. С. 4—28.
5. Куравский Л. С., Мармалюк П. А., Барабанщиков В. А., Безруких М. М., Демидов А. А., Иванов В. В., Юрьев Г. А. Оценка степени сформированности навыков и компетенций на основе вероятностных распределений глазодвигательной активности // Вопросы психологии. 2013. № 5. С. 64—81.
6. Марковские модели в задачах диагностики и прогнозирования: учеб. пособие / Под ред. Л. С. Куравского. М.: РУСАВИА, 2013. 172 с.
7. Овчаров Л. А. Прикладные задачи теории массового обслуживания. М.: Машиностроение, 1969. 324 с.
8. Крамер Г. Математические методы статистики. М.: Мир, 1976. 648 с.
9. Формалев В. Ф., Ревизников Д. Л. Численные методы. М.: ФИЗМАТЛИТ, 2004. 400 с.
10. Kuravsky L. S., Marmalyuk P. A., Yuryev G. A., Dumina P. N. A Numerical Technique for the Identification of Discrete-State Continuous-Time Markov Models // Applied Mathematical Sciences. 2015. Vol. 9. N. 8. P. 379—391.
11. Kuravsky L. S., Marmalyuk P. A., Baranov S. N., Alkhimov V. I., Yuryev G. A., Artyukhina S. V. A New Technique for Testing Professional Skills and Competencies and Examples of its Practical Applications // Applied Mathematical Sciences. 2015. Vol. 9. N. 21. P. 1003—1026.
12. The R Project for Statistical Computing. URL: <http://r-project.org/> (дата обращения: 25.04.2014).

L. S. Kuravsky, Professor, P. A. Marmalyuk, Professor Associated,
G. A. Yuryev, Professor Associated, P. N. Dumin, Graduate
Moscow State University of Psychology and Education, Computer Science Faculty,
127051 Moscow, Russia l.s.kuravsky@gmail.com

Methods of Numerical Identification for Markov Models and Their Comparative Analysis¹

Zero- and first-order numerical methods intended for identification of Markov processes with discrete states and continuous time using observation data as well as the computer experiment technology providing comparison of performance characteristics of the methods in question are under consideration. The first-order methods developed are characterized by utilizing both initial approximations derived from observation data and estimates of minimized criterion sensitivity to small variations of identified parameters. Analysis of the computer experiments' results showed the advantages of the developed methods over the classical first order gradient descent method. Improved running time of the methods is achieved through replacing, at each iteration, of enumeration of various combinations of parameter values in the neighborhood of their current estimates by enumeration of values of only those parameters to which the minimized criterion is highly sensitive. The algorithms under consideration can be used for tuning parameters of the trained structures including neural networks.

Keywords: Markov models, models identification, multivariate non-linear optimization, trained structures

References

1. Kuravskii L. S., Margolis A. A., Yur'ev G. A., Marmalyuk P. A. Kontsepsiya sistemy podderzhki prinyatiya reshenii dlya psikhologicheskogo testirovaniya, *Psikhologicheskaya nauka i obrazovanie*, 2012, no. 1, pp. 56–65.
2. Kuravskii L. S., Marmalyuk P. A., Alkhimov V. I., Yur'ev G. A. Matematicheskie osnovy novogo podkhoda k postroeniuyu protsedur testirovaniya, *Ekspierental'naya psikhologiya*, 2012. Vol. 5, no. 4, pp. 75–98.
3. Kuravskii L. S., Margolis A. A., Marmalyuk P. A., Yur'ev G. A., Dumin P. N. Obuchaemye markovskie modeli v zadachakh optimizatsii poryadka pred'yavleniya psikhologicheskikh testov, *Neirokomp'yutery: razrabotka i primeneniye*, 2013, no. 4, pp. 28–38.
4. Kuravskii L. S., Marmalyuk P. A., Alkhimov V. I., Yur'ev G. A. Novyi podkhod k postroeniuyu intellektual'nykh i kompetentnostnykh testov, *Modelirovaniye i analiz dannykh*, 2013, no. 1, pp. 4–28.
5. Kuravskii L. S., Marmalyuk P. A., Barabanshchikov V. A., Bezrukikh M. M., Demidov A. A., Ivanov V. V., Yur'ev G. A. Ot-senka stepeni sformirovannosti navykov i kompetentsii na osnove veroyatnostnykh raspredelenii glazodvigatel'noi aktivnosti, *Voprosy psikhologii*, 2013, no. 5, pp. 64–81.
6. Markovskie modeli v zadachakh diagnostiki i prognozirovaniya: Ucheb. posobie. Ed. L. S. Kuravskogo. Moscow, RUSAVIA, 2013, 172 p.
7. Ovcharov L. A. *Prikladnye zadachi teorii massovogo obsluzhivaniya*, Moscow, Mashinostroeniye, 1969. 324 p.
8. Kramer G. *Matematicheskie metody statistiki*. M.: Mir, 1976. 648 p.
9. Formalev V. F., Reviznikov D. L. *Chislennyye metody*. M.: FIZ-MATLIT, 2004, 400 p.
10. Kuravsky L. S., Marmalyuk P. A., Yuryev G. A., Dumin P. N. A Numerical Technique for the Identification of Discrete-State Continuous-Time Markov Models. *Applied Mathematical Sciences*, 2015, vol. 9, no. 8, pp. 379–391.
11. Kuravsky L. S., Marmalyuk P. A., Baranov S. N., Alkhimov V. I., Yuryev G. A., Artyukhina S. V. A New Technique for Testing Professional Skills and Competencies and Examples of its Practical Applications, *Applied Mathematical Sciences*. 2015, vol. 9, no. 21, pp. 1003–1026.
12. The R Project for Statistical Computing. URL: <http://r-project.org/> (дата обращения: 25.04.2014).

¹ This work was supported by the Russian Foundation for Basic Research (project № 14-06-00191) and the Russian Foundation for Humanities (project № 14-06-12012).

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.

Корректор *Е. В. Комиссарова*.

Сдано в набор 06.08.2015. Подписано в печать 23.09.2015. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ1015. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1.