

Международная федерация по обработке информации (IFIP)  
■ Федеральное агентство по образованию ■ Правительство  
Республики Карелия ■ Петрозаводский государственный  
университет ■ Национальный фонд подготовки кадров ■  
Государственный НИИ информационных технологий и теле-  
коммуникаций «Информика» ■ Российский фонд фундамен-  
тальных исследований ■ Американский благотворительный  
фонд поддержки информатизации науки и образования ■  
Политехнический музей (г. Москва) ■ Виртуальный компью-  
терный музей [www.computer-museum.ru](http://www.computer-museum.ru)

# **SORUCOM.2006**

## **РАЗВИТИЕ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ В РОССИИ И СТРАНАХ БЫВШЕГО СССР: ИСТОРИЯ И ПЕРСПЕКТИВЫ**

Материалы международной конференции  
(3–7 июля 2006 года)

В двух частях

*Часть 1*

Петрозаводск

2006

ББК 32.97  
УДК 681.3  
С 659

Редакционная коллегия:

Н.С. Рузанова (отв. редактор)

И.Г. Лежнев

С.А. Шлыкова

**SORUCOM.2006** : Развитие вычислительной техники в России  
С 659 и странах бывшего СССР: история и перспективы : материалы междунар.  
конф. (3–7 июля 2006 года) : В 2 ч. Ч. 1. – Петрозаводск, 2006. – 192 с.

ISBN 5-8021-0605-0

Сборник содержит материалы международной конференции, посвященной вопросам истории развития вычислительной техники в России и странах бывшего СССР, влияния и роли выдающихся ученых – руководителей различных школ и направлений развития вычислительной техники, архитектуры, структуры и конструкции отечественных вычислительных машин разных поколений и их программного обеспечения, организации разработки и применения вычислительных систем в различных регионах СССР и странах СНГ, вопросам образования в области информатики, освещения истории развития вычислительной техники в музеях, а также аспектам создания и использования современных IT-технологий.

ББК 32.97  
УДК 681.3

ISBN 5-8021-0605-0

© Петрозаводский государственный университет, 2006  
© Коллектив авторов, 2006

**ООО «МАЙКРОСОФТ РУС»**

Microsoft Rus, a Limited Liability Company

<http://www.microsoft.com/>

Основанная в 1975 году, корпорация Microsoft является мировым лидером в производстве программного обеспечения, предоставлении услуг и разработке Интернет-технологий для персональных компьютеров и серверов.

Корпорация Microsoft разрабатывает и выпускает широкий спектр программных продуктов. В их число входят настольные и сетевые операционные системы, серверные приложения для клиент-серверных сред, решения для управления бизнесом, настольные бизнес-приложения и офисные приложения для пользователей, интерактивные программы и игры, средства для работы в сети Интернет и инструменты разработки. Кроме того, Microsoft предлагает интерактивные (on-line) услуги, издает книги по компьютерной тематике, производит периферийное оборудование для компьютеров, занимается исследовательской деятельностью и разработкой новых компьютерных технологий. Продукты Microsoft продаются более чем в 80 странах мира, переведены более чем на 45 языков (в том числе на русский) и совместимы с большинством платформ персональных компьютеров.

С ноября 1992 года в России действует представительство Microsoft (с июля 2004 года – ООО «Майкрософт Рус»), в задачи которого входит продажа и продвижение программного обеспечения, развитие рынка информационных технологий, а также внедрение и локализация новейших технологий на территории России.

Microsoft стремится к максимальной реализации тех возможностей, которые открывают перед человеком информационные технологии. Компьютер на каждый стол и в каждый дом – эта концепция корпорации Microsoft сегодня дополнена новой концепцией Microsoft .NET (<http://www.microsoft.com/rus/net/>), отражающей стремление корпорации предоставить новейшие Интернет-технологии и Интернет-сервисы и расширить возможности человека за счет использования современного программного обеспечения в любое время, в любом месте и на любом устройстве.

Сегодня лицензионное программное обеспечение Microsoft используется на сотнях тысяч рабочих мест в России. На основе продуктов Microsoft ведутся проекты по внедрению мощных информационных систем в крупнейших отечественных коммерческих компаниях и государственных организациях. Более подробно о примерах внедрения ПО Microsoft в коммерческих компаниях: <http://www.microsoft.com/rus/business/>; в государственных организациях: <http://www.microsoft.com/rus/government/>.

Microsoft имеет все необходимые программные продукты для реализации концепции Microsoft .NET – надежные и масштабируемые серверные операционные системы и приложения Microsoft являются прекрасной платформой для предоставления Интернет-сервисов и организации эффективного бизнеса.

Дополнительную информацию о компании и продуктах Microsoft можно найти на веб-серверах Microsoft: <http://www.microsoft.com/rus> (рус. язык); <http://www.microsoft.com> (англ. язык).



## АМЕРИКАНСКИЙ БЛАГОТВОРИТЕЛЬНЫЙ ФОНД ПОДДЕРЖКИ ИНФОРМАТИЗАЦИИ НАУКИ И ОБРАЗОВАНИЯ

<http://www.inffond.ru>

В соответствии с уставом фонда его благотворительная деятельность может быть направлена на решение практически всего спектра задач в области информатизации образования и науки. Однако одновременно обеспечить такой широкий фронт даже при ограниченной благотворительной помощи по каждому направлению нереально в виду большой цены всего проекта. Поэтому в рамках существующих финансовых возможностей были определены следующие приоритеты:

- поддержка наиболее авторитетных (в масштабах страны) конференций, семинаров и «крутых столов» в области информатизации образования и науки;
- адресная помощь коллективам и авторам наиболее интересных и значимых проектов в области информатизации;
- поддержка мероприятий и проектов в разных, прежде всего удаленных, регионах страны;
- благотворительная помощь в информатизации образования для людей с ограниченными физическими и умственными возможностями;
- поддержка конкретных специалистов в области информатизации в сфере образования и науки, добившихся большого интегрального вклада в решение проблемы.



## ASSOCIATION FOR COMPUTING MACHINERY (ACM)

<http://www.acm.org>

ACM, the Association for Computing Machinery, is an international scientific and educational organization dedicated to advancing the arts, sciences, and applications of information technology. With a world-wide membership ACM is a leading resource for computing professionals and students working in the various fields of Information Technology, and for interpreting the impact of information technology on society.

## ACM SPECIAL INTEREST GROUP ON COMPUTER SCIENCE EDUCATION (SIGCSE)

<http://www.sigcse.org>

The ACM Special Interest Group on Computer Science Education provides a forum for problems common among educators working to develop, implement and/or evaluate computing programs, curricula, and courses, as well as syllabi, laboratories, and other elements of teaching and pedagogy.

**INTERNATIONAL FEDERATION FOR  
INFORMATION PROCESSING (IFIP)**

*<http://www.ifip.or.at>*

IFIP is a non-governmental, non-profit umbrella organization for national societies working in the field of information processing. It was established in 1960 under the auspices of UNESCO as an aftermath of the first World Computer Congress held in Paris in 1959. Today, IFIP has several types of Members and maintains friendly connections to specialized agencies of the UN system and non-governmental organizations. Technical work, which is the heart of IFIP's activity, is managed by a series of Technical Committees

IFIP enjoys friendly cooperation with a number of international organisations. First among these is UNESCO with which IFIP maintains a formal consultative relationship. UNESCO has commissioned from IFIP several projects and supports the participation of some people from developing countries in IFIP events. IFIP is a Scientific Associate of ICSU (International Council of Scientific Unions).

There are four international federations with which IFIP collaborates. These are IFAC (International Federation of Automatic Control), near neighbours of ours in Laxenburg, IMACS (International Association for Mathematics and Computers in Simulation), IFORS (International Federation of Operational Research Societies) and IMEKO (International Measurement Confederation)..

**CISCO SYSTEMS**

*<http://www.cisco.ru>*

Cisco Systems – мировой лидер в области сетевых технологий, предназначенных для сети Интернет.

Решения Cisco объединяют людей, компьютерные устройства и компьютерные сети и позволяют людям получать и передавать информацию, независимо от места, времени и используемых компьютерных систем.

Cisco разрабатывает комплексные решения, с помощью которых заказчики создают собственные объединенные информационные инфраструктуры или получают доступ к сетям других владельцев. При этом комплексным мы называем такое решение, которое создает общую архитектуру для оказания согласованных сетевых услуг всем абонентам. Чем шире спектр сетевых услуг, тем полезнее для подключенных абонентов будет данная сеть.

Наши ресурсы – это самый широкий в отрасли портфель аппаратных средств, которые используются для строительства информационных сетей и предоставления доступа к ним; операционная система Cisco IOS, предназначенная для поддержки сетевых услуг и приложений; опыт проектирования и развертывания сетей; а также система технической поддержки и профессиональных услуг по обслуживанию и оптимизации сетевой работы.



## ООО «ИБМ ВОСТОЧНАЯ ЕВРОПА/АЗИЯ»

IBM East Europe/Asia

<http://www.ibm.com>

Корпорация International Business Machines является одним из ведущих мировых поставщиков аппаратных и программных решений, лидером по разработке и производству информационных технологий и устройств хранения данных. Уже на протяжении 13 лет IBM лидирует по количеству патентов в мире, за прошлый год IBM получила 2941 патент. По результатам исследования компании IDC мирового серверного рынка в 2005 году, IBM является компанией № 1 среди западных производителей с долей рынка 32,9%. IBM занимает первое место по количеству суперкомпьютеров в мире. В лабораториях IBM разрабатываются технологии, которые впоследствии становятся стандартом для развития компьютерной индустрии.

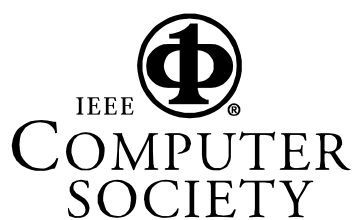
*IBM Global Services* является крупнейшим подразделением IBM. В настоящее время Global Services приносит около половины годового дохода IBM и является наиболее динамично развивающимся бизнесом в составе корпорации.

*IBM Business Consulting Services* является крупнейшей в мире консалтинговой организацией. IBM Business Consulting Services предоставляет клиентам свой опыт и экспертизу в области построения бизнес-процессов, глубокое понимание различных отраслей и специфических индустриальных задач и умение разработать, построить и внедрить решения, обеспечивающие повышение общей эффективности деятельности компаний.

В России корпорация IBM активно работает с 1974 года. IBM стала одной из первых зарубежных компаний, которая зарегистрировала в 1991 году свой филиал как советское (а затем российское) юридическое лицо со 100%-ным иностранным капиталом. В 1993 году компания IBM СССР была переименована в IBM East Europe/Asia. Также IBM представлена в активно развивающихся странах СНГ, таких как Украина, Казахстан и Узбекистан.

Около 500 бизнес-партнеров активно занимаются продвижением широкого спектра решений на базе технологий IBM в России и странах СНГ. В IBM существует специальное подразделение, основной задачей которого является эффективная работа с поставщиками решений, независимыми разработчиками программного обеспечения и системными интеграторами в России и странах СНГ. Такое всестороннее сотрудничество позволяет активно предлагать интегрированные решения в управлении ресурсами предприятия, цепочкой поставок и других областях, которые востребованы заказчиками. Привлекательность сотрудничества с IBM для клиентов и бизнес-партнеров подкрепляется наличием экспертизы IBM в области информационных технологий и бизнес-консалтинга. Тысячи коммерческих и государственных организаций выбрали технологии IBM для построения ИТ-инфраструктуры, организации бизнес-процессов и повышения эффективности работы.

Поддержка перспективных инновационных технологий и научные исследования позволяют корпорации IBM сохранять свое лидерство в современных сложных условиях рынка информационных технологий.

**IEEE COMPUTER SOCIETY**

*<http://computer.org>*

The IEEE Computer Society traces its origins to the 1946 formation of the Subcommittee on Large-Scale Computing of the American Institute of Electrical Engineers (AIEE). Five years later, the Institute of Radio Engineers (IRE) formed its Professional Group on Electronic Computers. The principal volunteer officers of both these groups were designated chairs. The AIEE and the IRE merged in 1963 to become the Institute of Electrical and Electronics Engineers (IEEE). The respective committee and group of the predecessor organizations combined to form the modern IEEE Computer Society. The society's principal volunteer officer has been designated as president since 1970. The Computer Society celebrates its fiftieth anniversary year in 1996.

With nearly 100,000 members, the IEEE Computer Society is the world's leading organization of computer professionals. Founded in 1946, it is the largest of the 39 societies of the IEEE.

The IEEE Computer Society's vision is to be the leading provider of technical information, community services, and personalized services to the world's computing professionals.

The Society is dedicated to advancing the theory, practice, and application of computer and information processing technology. Through its conferences, applications-related and research-oriented journals, local and student chapters, distance learning campus, technical committees, and standards working groups, the Society promotes an active exchange of information, ideas, and technological innovation among its members. In addition, the Society maintains close ties with the US Computing Sciences Accreditation Board and Accreditation Board for Engineering and Technology, monitoring and evaluating curriculum accreditation guidelines.

With about 40 percent of our members living and working outside the United States, the Computer Society fosters international communication, cooperation, and information exchange. To meet the needs of our members conveniently and efficiently, the Society maintains a service center office in Tokyo, Japan; a publications office in Los Alamitos, California; and its headquarters in Washington, DC

## ИНФОРМАЦИОННЫЕ СПОНСОРЫ



### Информационное агентство связи

127091, Москва, Делегатская ул., 5а. Тел./Факс: (495) 337-02-22

E-mail: [info@operato.ru](mailto:info@operato.ru), <http://www.operato.ru>

Представляет еженедельный информационно-справочный бюллетень «Оператор. Новости связи», выходит в электронном и бумажном виде. Это – структурированные новости от всех участников рынка телекоммуникаций России и зарубежных стран, аналитическая и справочная информация по рынку инфокоммуникаций. Приводится реакция СМИ на события за неделю.

Цель издания – стать «коммутатором» отраслевой информации.



### CNEWS

117393, Москва, Профсоюзная ул., 78, РБК, CNews.ru. Тел.: (495) 363-1157

E-mail: [news@cnews.ru](mailto:news@cnews.ru), <http://www.cnews.ru>

Издание основано в 2000 году в рамках медиа-проектов холдинга РБК. На данный момент в издании ежедневно публикуется более 100 сообщений об ИТ-рынке России и мира. В поле зрения новостного отдела CNews.ru находятся рынки телекоммуникаций, компьютерного оборудования, программного обеспечения, электронного бизнеса и защиты информации. Информация, опубликованная на [www.cnews.ru](http://www.cnews.ru), опережает публикации в других средствах массовой информации.

Ежедневно более 75 тысяч читателей посещают наш сайт. Ежемесячная аудитория составляет более чем 800 тысяч человек по всей России и за рубежом. Основу аудитории издания составляют руководители, менеджеры и квалифицированные специалисты промышленных и ИТ-компаний, банков и государственных учреждений.

С 2004 года начал выходить журнал CNews.



Информационные ресурсы России – <http://www.kcni.ru/irr/>



### CITforum

ИТ-портал – крупнейшее российское медиапространство, ориентированное на ИТ-профессионалов. Суммарная месячная аудитория – 500 тысяч человек.

Головной проект – <http://citforum.ru>, крупнейшая техническая библиотека материалов по компьютерным технологиям. Библиотека собирается с 1997 года и охватывает тематику инженерии программного обеспечения и программирования, ОС, средств управления данными, сетевых и Интернет-технологий и т.д. В ИТ-портал CITforum.ru входит еще 2 крупных ресурса:

**CITKIT.ru** – крупнейший в России архив свободно распространяемых программ. Объем архива – более 4 терабайт. Это единственный в мире крупный архив, снабженный собственным веб-каталогом;

**CitCity.ru** – деловая on-line газета, посвященная корпоративным применениям информационных технологий.

Основной контент проекта – новости. Периодически публикуются статьи аналитического и обзорного характера. Также портал включает несколько небольших проектов, в их числе форумы (<http://forum.citforum.ru>), литературно-развлекательный проект (<http://kurilka.citforum.ru>) и др.



### Компания Begin Group

Проект BEGIN – [www.begin.ru](http://www.begin.ru) – все об экономическом и бизнес-образовании! Подробная информация о бизнес-школах и программах MBA, о презентациях бизнес-школ и тематических выставках, материалы, посвященные вопросам подготовки и поступления на программы MBA, активные форумы.

Тел.: (495) 933-1098, E-mail: [webmaster@begin.ru](mailto:webmaster@begin.ru)

Проект HRM – [www.hrm.ru](http://www.hrm.ru) – все о кадровом менеджменте! Доступ к базам данных тренинговых и консалтинговых компаний, кадровых агентств, компаний, предлагающих услуги по формированию социального пакета, новости о специализированных конференциях и выставках, большая подборка тематических статей и активные форумы по различным вопросам управления персоналом.

Тел.: (495) 933-1098. E-mail: [hrm@hrm.ru](mailto:hrm@hrm.ru)

## МЕЖДУНАРОДНЫЙ ПРОГРАММНЫЙ КОМИТЕТ

**Я.А. Хетагуров**

*сопредседатель МПК, д.т.н., проф.  
Россия, Москва*

**В.В. Пржиялковский**

*сопредседатель МПК, д.т.н.,  
НИИ «Аргон», консультант  
Россия, Москва*

**А.Н. Тихонов**

*директор ГНИИ ИТТ «Информика», проф., д.т.н., чл.-корр.  
РАО  
Россия, Москва*

**А.В. Воронин**

*ректор ПетрГУ, проф., д.т.н.  
Россия, Петрозаводск*

**И.Р. Агамирзян**

*директор по стратегическому развитию Microsoft Россия и СНГ  
Россия, Москва*

**Т.М. Александриди**

*проф. кафедры АСУ МАДИ (ГТУ)  
Россия, Москва*

**Ю.А. Богоявленский**

*к.т.н, зав. кафедрой ПетрГУ  
Россия, Петрозаводск*

**Н.П. Брусенцов**

*МГУ, зав. лабораторией, член Совета Виртуального компьютерного музея [www.computer-museum.ru](http://www.computer-museum.ru)  
Россия, Москва*

**В.Г. Домрачев**

*проф., д.т.н., президент Американского благотворительного фонда поддержки информатизации образования и науки «Информатизация»  
Россия, Москва*

**М. Згуровский**

*National Technical University  
of Ukraine  
Украина*

**В.П. Иванников**

*Институт системного  
программирования РАН  
Россия, Москва*

**Дж. Импаглияццо**

*Hofstra University  
США*

**Ю.В. Карпилович**

*к.т.н, проф., лауреат Ленинской  
и Государственной премий СССР, ве-  
дущий специалист ОАО «МПОБТ»  
Беларусь*

**Б.Н. Малиновский**

*член-корр. АН Украины,  
Институт кибернетики НАНУ,  
д.т.н., проф.  
Украина*

**Н.Л. Прохоров**

*генеральный директор ИНЭУМа  
Россия, Москва*

**Г.А. Оганян**

*д.т.н., проф., Армения*

**Э.М. Пройдаков**

*редакционный директор ИТ-изданий  
«СК Пресс», директор виртуального  
компьютерного музея  
Россия, Москва*

**Ю.В. Рогачев**

*д.т.н. Россия, Москва*

**М.Э. Смолевицкая**

*с.н.с., Политехнический музей  
Россия, Москва*

**А.Н. Томилин**

*д.физ.-мат.н., проф., Институт  
системного программирования  
РАН, нач. отдела  
Россия, Москва*

**А. Фицпатрик**

*Federation of American Scientists  
США*

## ОРГАНИЗАЦИОННЫЙ КОМИТЕТ

**В.Н. Васильев**

*председатель, президент ПетрГУ,  
проф., д.т.н.  
Россия, Петрозаводск*

**Н.С. Рузанова**

*заместитель председателя,  
профессор ПетрГУ, директор  
РЦНИИТ, советник Главы  
Республики Карелия  
Россия, Петрозаводск*

**Джон Импаглияццо**

*Hofstra University, США*

**О.Ю. Насадкина**

*Россия, Петрозаводск*

**С.А. Шлыкова**

*Россия, Петрозаводск*

**А.В. Голубев**

*Россия, Петрозаводск*

## АВТОМАТИЧЕСКАЯ ЦИФРОВАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА М-1 (АЦВМ М-1)

**Т.М. Александриди, Ю.В. Рогачев**

МАДИ (ГТУ)

Москва, РФ

*alexandridi@mail.ru*

В декабре 1951 года в лаборатории электросистем Энергетического института АН СССР под руководством члена-корреспондента АН СССР Исаака Семеновича Брука был выпущен научно-технический отчет «Автоматическая цифровая вычислительная машина [М-1]», утвержденный 15 декабря 1951 года директором ЭНИН АН СССР академиком Г.М. Кржижановским. Это был первый в СССР научный документ о создании отечественной ЭВМ. Машина успешно прошла испытания и была переведена в режим эксплуатации для решения задач как в интересах институтов Академии наук СССР, так и других организаций.

Начало исследовательских работ И.С. Брука по проблеме цифровых вычислительных машин относится к 1948 году. Он первым в СССР (совместно с Б.И. Рамеевым) разработал проект цифровой ЭВМ с жестким программным управлением. Первое в СССР свидетельство об изобретении на «ЦВМ с общей шиной» было получено ими в декабре 1948 года.

Постановление Президиума АН СССР о начале разработки М-1 вышло 22 апреля 1950 года. Брук получил возможность формировать команду разработчиков. Первым был принят Николай Яковлевич Матюхин, молодой специалист, только что окончивший радиотехнический факультет Московского энергетического института. Брук ознакомил его с основными идеями построения цифровой вычислительной машины, и затем они вместе разрабатывали структуру и состав будущей машины, ее основные характеристики и конкретные решения многих технических вопросов. В дальнейшем Н.Я. Матюхин при постоянной поддержке И.С. Брука практически выполнял функции главного конструктора.

Вот отрывок из воспоминаний Николая Яковлевича:

«Я хочу постараться оживить картинки нашей работы под руководством Исаака Семеновича, передать атмосферу первых лет работ в области вычислительной техники.

Формирование группы и начало работ над АЦВМ М-1 – 1950 год.

Брук набирает на РТФ МЭИ команду молодых специалистов, нас семеро: два младших научных сотрудника – Залкинд А.Б., Матюхин Н.Я.; два дипломника – Александриди Т.М., Карцев М.А.; три техника – Рогачев Ю.В., Шидловский Р.П., Журкин Л.М.

Первое задание Исаака Семеновича мне – построить ламповый диодный трехходовой сумматор (проверка моей пригодности).

Второе задание – спроектировать типовой рабочий стол.

Третье задание мне как руководителю группы – разработка АЦВМ М-1.

Серьезные трудности при проектировании М-1 и реализации проекта создавало почти полное отсутствие комплектующих изделий. Исаак Семенович нашел оригинальный выход, используя имущество со складов военных трофеев. В результате в основу проекта были положены следующие идеи и «трофеи»:

- сочетание малой номенклатуры компонент самого разного происхождения;
- всего два типа электронных ламп;
- купроксные выпрямители электроизмерительных приборов;
- магнитные головки от бытового магнитофона;

- электронно-лучевые трубки от осциллографа;
  - трофейный телетайп из Генштаба Вермахта.
- О стиле руководства Исаака Семеновича Брука:
- глубокое понимание цели, простота и образность аргументации;
  - никаких «разносов» по неудачам;
  - уважительное отношение к исполнителям.

Занимаясь созданием машины М-1, приходилось разбираться в самых разных вопросах – от регулятора напряжения для мощных мотор-генераторов постоянного тока, служивших источниками вторичного питания машины, до разработки системы команд и программирования первых задач. Предстояло разработать архитектуру машины, ее основные устройства и систему элементов с учетом указанных выше ограничений в обеспечении комплектующими изделиями.

АЦВМ М-1 включала в свой состав арифметическое устройство, главный программный датчик (устройство управления), внутреннюю память двух видов (быструю – на электростатических трубках и медленную – на магнитном барабане), устройство ввода-вывода с использованием телеграфной буквопечатающей аппаратуры.

Осенью 1950 года «команда» Н.Я. Матюхина была укомплектована.

Разработка арифметического устройства и системы логических элементов выполнялась Н.Я. Матюхиным и Ю.В. Рогачевым, разработка главного программного датчика – М.А. Карцевым и Р.П. Шидловским, запоминающего устройства на магнитном барабане – Н.Я. Матюхиным и Л.М. Журкиным, запоминающего устройства на электростатических трубках – Т.М. Александриди, устройства ввода-вывода – А.Б.Залкиндом и Д.У. Ермоченковым, разработка электропитания – В.В. Бельским, разработка конструкции – А.И. Кокалевским. Комплексную отладку машины и отработку технологии программирования и тестирования возглавил Н.Я. Матюхин. Все разработчики проявляли исключительную изобретательность, приспособлявая к использованию в ЦВМ комплектующие изделия, ранее предназначавшиеся для других целей (магнитные головки от бытовых магнитофонов, электронно-лучевые трубки от осциллографов, буквопечатающий телетайп и др.).

В процессе проектирования и разработки М-1 были предложены и реализованы принципиально новые технические решения, в частности двухадресная система команд, нашедшая впоследствии широкое применение в отечественной и зарубежной вычислительной технике. Вспоминая позднее об этом решении, Н.Я. Матюхин писал: «Сам выбор системы команд был для нас делом непростым – в то время общепринятой и наиболее естественной считалась трехадресная система, шедшая еще от работ фон Неймана, которая требовала достаточно большой разрядности регистрового оборудования и памяти. Наши ограниченные возможности стимулировали поиск более экономных решений. Как иногда бывает в тупиковых ситуациях, помог случай. Брук в то время пригласил на работу молодого математика Ю.А. Шрейдера. Шрейдер, осваивая вместе с нами азы программирования, обратил наше внимание на то, что во многих формулах приближенных вычислений результат операции становится для следующего шага одним из операндов. Отсюда было уже недалеко до первой двухадресной системы команд. Наши предложения были одобрены Бруком, и АЦВМ М-1 стала первой двухадресной машиной».

Впервые в мировой практике создания цифровых электронных вычислительных машин в М-1 диодные логические схемы строились на полупроводниковых элементах. Первоначально разработка начиналась на ламповых диодах 6Х6. Был изготовлен с этими лампами и отлаживался макет двоичного счетчика. Но помогло случайное сочетание обстоятельств. Поиски путей сокращения количества радиоламп в машине привели к попытке использовать вместо радиоламп 6Х6 купроксные выпрямители КВМП-2-7, оказавшиеся на складе лаборатории среди трофейного имущества.

**Параметры купроксного выпрямителя КВМП-2-7**

|                                      |           |
|--------------------------------------|-----------|
| Допустимый прямой ток .....          | 4 ма      |
| Прямое сопротивление .....           | 3–5 Ком   |
| Допустимое обратное напряжение ..... | 120 в     |
| Обратное сопротивление.....          | 0,5–2 Мом |

Соотношение прямого и обратного сопротивлений (не хуже, чем 1:100) надежно могло обеспечить выполнение диодных функций в логических схемах. Предстояло только выяснить возможность их использования в импульсных схемах при работе на достаточно высокой частоте. Экспериментальные исследования логических схем, разработанных с учетом этих параметров, показывали их стабильную работу. Были проведены исследования работы макета одного разряда арифметического устройства с сумматором, дешифраторами и смесителями на выпрямителях КВМП-2-7 в различных режимах и с учетом отклонений уровней питающих напряжений и разбросов параметров комплектующих изделий. Макет работал устойчиво. В августе 1950 года были проведены его заключительные испытания с непосредственным участием И.С. Брука, которые также подтвердили надежную работу логических схем, построенных на базе купроксных выпрямителей КВМП-2-7. Бруком было принято решение в пользу этого варианта.

В своих воспоминаниях Н.Я. Матюхин так оценивал значение этого решения: «Одним из принципиальных решений, которое, как мне кажется, предопределило успех нашей первой машины и короткие сроки ее создания, был курс, принятый Бруком на широкое использование полупроводниковых элементов. Тогда они были представлены в нашей промышленности только малогабаритными купроксными выпрямителями, которые выпускались для нужд измерительной техники. Брук договорился о выпуске специальной модификации такого выпрямителя размером с обычное сопротивление, и мы создали набор типовых схем. В мастерской при лаборатории началось изготовление и монтаж блоков, и менее чем через год машина уже «задышала» (а было в ней несколько сотен ламп и несколько тысяч купроксов). Так АЦВМ М-1 стала первой в мире цифровой вычислительной машиной, в которой логические схемы строились на полупроводниковых приборах.

Серьезные проблемы возникли при проектировании электростатической памяти.

Известно, что в США и в СССР для этой цели разрабатывались специальные электронно-лучевые трубки – потенциалоскопы. Однако в СССР эти трубки предназначались для тех организаций, которые разрабатывали ЭВМ по постановлениям правительства. К сожалению, на потенциалоскопы мы не могли рассчитывать, так как И.С. Брук вел разработку ЦВМ как инициативные исследования по планам АН СССР. Тогда он решил использовать для этой цели обычные электронно-лучевые трубки ЛО-737, которые применялись в осциллографах.

Примерно в это же время в сентябре 1950 года в лаборатории появилась новая дипломница радиофакультета Московского энергетического института Тамара Александровна.

И.С. Брук предложил ей выполнить дипломный проект на тему «Электростатическое запоминающее устройство», которое предназначалось для АЦВМ-1.

Сначала исследования проводились непосредственно на обычном осциллографе, укомплектованном трубками ЛО-737 и дополнительными измерительными схемами.

Уже первые эксперименты показали, что эффект запоминания информации на экране электронно-лучевой трубки может быть реализован. Сразу же началось проектирование электростатического запоминающего устройства для М-1.

Рассмотрим кратко физические процессы, на которых основана электростатическая память (ЭП). Информация в памяти хранится в виде определенного распределения статических зарядов на экране электронно-лучевой трубки. Эффект запоминания основан на явлении вторично-электронной эмиссии. При опреде-

ленной величине ускоряющего напряжения коэффициент вторичной эмиссии экрана больше единицы, т.е. при бомбардировке экрана лучом число вторичных электронов, покидающих экран, больше числа первичных электронов, попадающих на него. Вследствие этого облучаемый участок экрана приобретает положительный заряд. Для записи двоичной информации использовалась система чтения-записи «фокус-дефокус», при которой «1» записывалась сфокусированным лучом, «0» – расфокусированным лучом, считывание выполнялось расфокусированным лучом. При считывании «1» появляется положительный сигнал, но при этом информация стирается. Поэтому после чтения выполняется регенерация, т.е. снова записывается «1».

Весной 1951 года Т. Александриди под руководством И.С. Брука написала и защитила диплом инженера на тему «Электростатическое запоминающее устройство».

К этому же времени в лаборатории была изготовлена ЭП и началась ее наладка.

К концу 1951 года впервые в мире в опытную эксплуатацию поступила ЦВМ-1 с быстродействующим запоминающим устройством.

Конструктивно ЭП состояла из 8 электронно-лучевых трубок ЛО-737 и блоков развертки и управления.

На экране каждой трубки находились 32 строки, в каждой из которых содержались 25 точек, т.е. одно число или команда. Следовательно, общий объем быстродействующей ЭП был равен 256 25-разрядных двоичных слов. Операция сложения с ЭП выполнялась за 50 мксек, т.е. обеспечивалось быстродействие 20 000 операций в секунду.

Осенью 1951 года закончились работы по настройке машины М-1. К декабрю этого же года машина успешно прошла комплексные испытания и была передана в эксплуатацию.

### **Основные характеристики М-1**

Система счисления .....двоичная  
Количество двоичных разрядов .....25  
Система команд ..... двухадресная

Объем внутренней памяти:  
на электростатических трубках .....256 адресов  
на магнитном барабане .....256 адресов

Быстродействие:  
с медленной памятью .....20 оп/сек  
с быстрой памятью  
операция сложения .....за 50 мксек  
операция умножения .....за 2000 мксек

Количество электронных ламп .....730  
Потребляемая мощность .....8 Квт  
Занимаемая площадь .....4 кв. метра

Ознакомиться с работой АЦВМ М-1 приезжали видные ученые, в том числе академики А.Н. Несмеянов, М.А. Лаврентьев, С.Л. Соболев, А.И. Берг. Одним из первых на М-1 решал задачи по ядерным исследованиям академик С.Л. Соболев, бывший в то время заместителем по научной работе в институте И.В. Курчатова.

Три года машина М-1 находилась в эксплуатации, при этом первые полтора года была единственной в Российской Федерации действующей ЭВМ. Она была изготовлена в единственном экземпляре, но ее архитектура и многие принципиальные схемные решения были приняты в дальнейшем за основу при разработке серийных машин М-3, «МИНСК», «РАЗДАН» и др.

Создатели машины М-1 стали крупными специалистами в области вычислительной техники и внесли значительный вклад в ее развитие, возглавляя различные научные, учебные и производственные коллективы. Их труд был высоко оценен присвоением ученых степеней и почетных званий, присуждением государственных наград.

АКАДЕМИЯ НАУК СССР

Инв. № 683

ЭНЕРГЕТИЧЕСКИЙ ИНСТИТУТ им. Г.М. КРЖИЖАНОВСКОГО.

Лаборатория ЭЛЕКТРОСИСТЕМ.

ОТЧЕТ ПО РАБОТЕ: АВТОМАТИЧЕСКАЯ ЦИФРОВАЯ ВЫЧИСЛИТЕЛЬНАЯ  
МАШИНА.

[М-1]

Директор Энергетического ин-та  
АН СССР  
академик

Академия Наук СССР  
ИНСТИТУТ  
ЭЛЕКТРОННЫХ УСТРОЙСТВ  
Москва Б-2, Ленинский проспект д. 19

Г.М. Кржижановский/

Руководитель лаборатории  
Электросистем  
Член. корр. АН СССР

И.С. Брук/

Исполнители работы  
Младшие научные сотр.

Александр /Т.М. Александриди/

Залкин /А.Б. Залкин/

Карцев /М.А. Карцев/

Матюхин /Н.Я. Матюхин/

Журикин /П.М. Журикин/

Рогов /Ю.В. Рогов/

Шидловский /Р.П. Шидловский/

Техники:

А Н Н О Т А Ц И Я

В отчете дается краткое описание построенной  
машины и принцип действия отдельных ее устройств.

Москва  
1951г.

№1539  
15/44-51г  
3 экз

## ЭВМ М-2

**Т.М. Александриди, Ю.А. Лавренюк, Ю.В. Рогачев**

МАДИ (ГТУ)

Москва, РФ

*alexandridi@mail.ru*

Научный руководитель член-корреспондент АН СССР И.С. Брук.

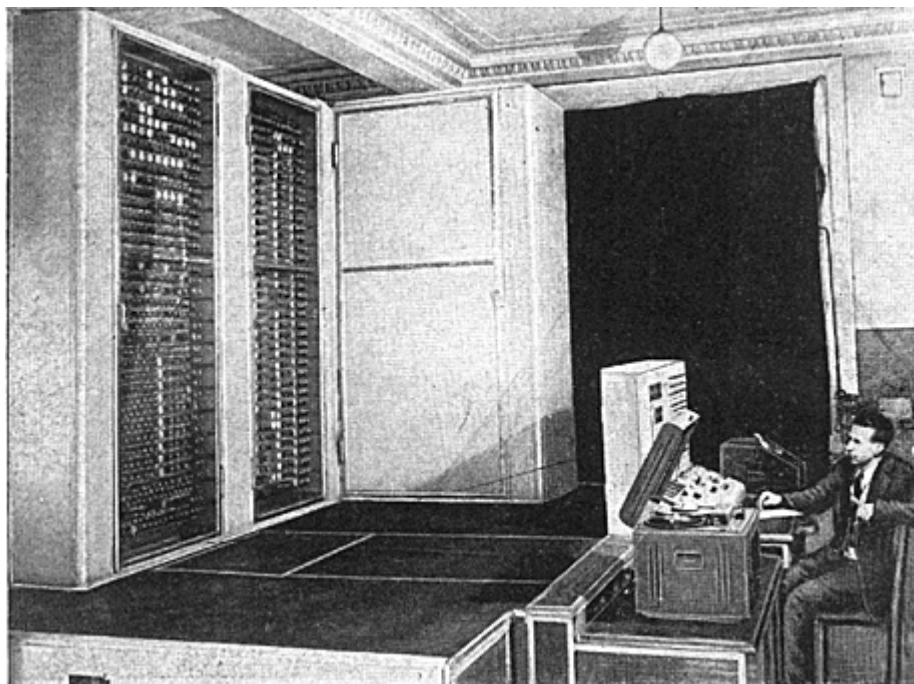
Руководитель группы разработчиков М.А. Карцев.

В группу, работавшую над М-2, входили на разных этапах работы от 7 до 10 инженеров. Арифметический узел разрабатывался М.А. Карцевым, В.В. Белынским и А.Б. Залкиндом, электростатическое запоминающее устройство – Т.М. Александриди и Ю.А. Лавренюком, устройство управления – Л.С. Легезо, В.Д. Князевым и Г.И. Танетовым, магнитные запоминающие устройства – А.И. Щуровым и Л.С. Легезо, входные и выходные устройства – А.Б. Залкиндом, система питания – В.В. Белынским, Ю.А. Лавренюком и В.Д. Князевым, пульт управления – В.В. Белынским и А.И. Щуровым.

Большая работа проведена конструкторами, техниками, механиками и монтажниками лаборатории – И.З. Гельфгатом, А.Д. Гречушкиным, Н.А. Немцевым, Ф.Ф. Ржеутским, И.К. Швильпе, Д.У. Ермоченковым, Л.И. Федоровым, Г.В. Коростылевым и др.

ЭВМ М-2 разработана и изготовлена в единственном экземпляре в лаборатории электросистем Энергетического института АН СССР им. Г.М. Кржижановского по постановлению Президиума Академии наук СССР. В состав машины входили устройства: арифметический узел, программный датчик (устройство управления), внутренние запоминающие устройства – устройства ввода и вывода данных, внешнее запоминающее устройство.

Разработка и монтаж машины были проведены в весьма короткий срок – с апреля по декабрь 1952 года. В январе 1953 года началась эксплуатация ЭВМ М-2 с памятью на магнитном барабане. В июне было подключено и электростатическое запоминающее устройство на электронно-лучевых трубках 13-ЛО-37.



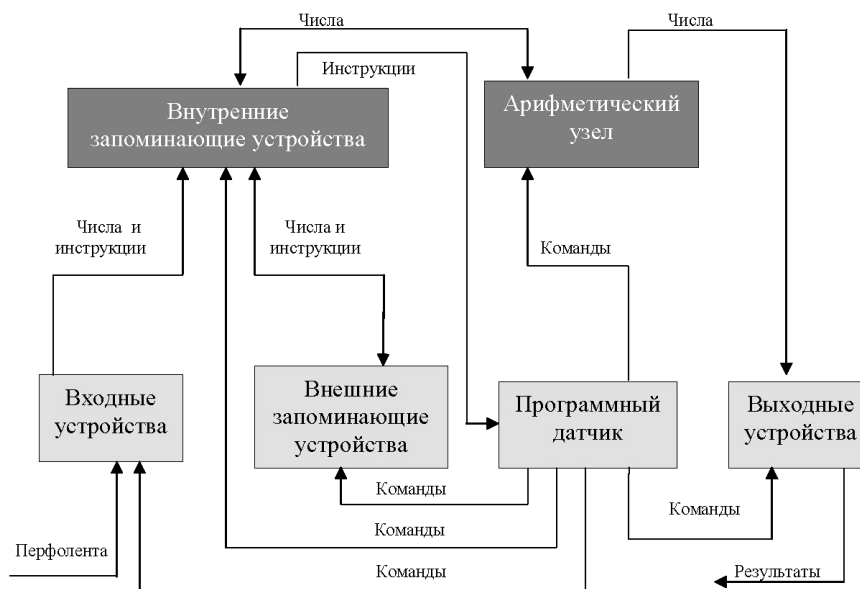
*Общий вид вычислительной машины М-2*

### Основные характеристики М-2

|                                      |                                                                                                                                                 |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Система счисления .....              | двоичная                                                                                                                                        |
| Представление чисел .....            | с плавающей запятой и с фиксированной запятой                                                                                                   |
| Количество двоичных разрядов .....   | 34                                                                                                                                              |
| Система кодирования инструкций ..... | трехадресная                                                                                                                                    |
| Выполняемые операции .....           | сложение, вычитание, умножение, деление,<br>сравнение по модулю, сравнение алгебраическое,<br>логическое умножение и др.<br>(всего 30 операций) |
| Быстродействие .....                 | 2000 – 3000 оп/с.                                                                                                                               |
| Внутренние ЗУ                        |                                                                                                                                                 |
| магнитный барабан .....              | 512 34-разрядных двоичных чисел                                                                                                                 |
| электростатическое .....             | 512 34-разрядных двоичных чисел                                                                                                                 |
| ферритовое .....                     | 4096 34-разрядных двоичных чисел                                                                                                                |
| Внешнее ЗУ .....                     | магнитная лента на 50 тыс. чисел                                                                                                                |
| Ввод данных .....                    | трансмиссером с перфорированной бумажной ленты<br>со скоростью 30 чисел в секунду                                                               |
| Вывод данных .....                   | в виде таблиц, печатаемых рулонным телетайпом.<br>Скорость печати 24 числа в минуту                                                             |

Зимой 1954/55 года машина была существенно модернизирована. В целях ее дальнейшего усовершенствования в 1956 году было разработано, изготовлено и введено в состав машины ферритовое оперативное запоминающее устройство емкостью 4096 слов. Это потребовало введения специального регистра для запоминания того, какая область памяти используется в данный момент времени, и специальной операции изменения содержимого этого регистра (переключение областей памяти). В машине М-2 впервые была реализована идея укороченных адресов в командах и укороченных кодов операций как способа согласования форматов команд и форматов чисел. Эта идея была предшественником способов формирования исполнительных адресов в машинах второго и третьего поколений.

### Блок-схема М-2



### Основные узлы машины

*Арифметический узел (АУ)* предназначен для производства операций над числами. В нем выполняются сложение, вычитание, умножение, деление, а также некоторые логические операции.

В АУ имеются три основных регистра, обозначаемых буквами А, В и С, и вспомогательный регистр Е. При сложении и вычитании числа помещаются в регистры В и С, результат получается в регистре В. При умножении сомножители помещаются в регистры А и С, а в регистре В образуются частные суммы и произведение. При выполнении деления делимое помещается в регистр В, делитель – в регистр С, а в регистре А образуется частное.

Все числа принимаются из запоминающих устройств в регистр А и затем передаются в другие регистры АУ. Запись результата операции в запоминающее устройство также производится из регистра А. Таким образом, регистр А служит связующим звеном между внутренними ЗУ и арифметическим узлом. Этот же регистр служит для связи внутренних запоминающих устройств с внешним запоминающим устройством и с устройствами ввода и вывода.

Каждый из регистров состоит из 33 триггеров, занумерованных от 1 до 33.

В зависимости от того, в какой системе ведется счет (с плавающей или с фиксированной запятой), работа регистров АУ организуется различным образом. При действиях с фиксированной запятой регистры работают как единое целое, и каждый триггер регистра соответствует определенному разряду числа. В этом случае триггер № 1 регистра представляет собой старший двоичный разряд числа.

Если выполняются действия с плавающей запятой, то регистры делятся на две части: в триггерах с № 1 по № 6 устанавливается порядок числа, а в триггерах с № 7 по № 33 – мантисса числа. В этом случае триггер № 1 регистра определяет знак порядка, а триггер № 7 является старшим разрядом мантиссы.

Арифметический узел полностью параллельного типа: прием чисел из запоминающих устройств, передача их из одного регистра в другой и суммирование ведутся одновременно по всем разрядам.

*Запоминающие устройства.* В ЭВМ М-2 имелось два внутренних запоминающих устройства: магнитное и электростатическое. Оба запоминающих устройства параллельного типа: запись и чтение ведутся одновременно по всем разрядам.

В магнитном запоминающем устройстве каждая цифра запоминаемого числа представляется в виде миниатюрного магнетика одного или другого направления, получающегося при намагничивании участка поверхности вращающегося барабана. Барабан представляет собой дюралюминиевый цилиндр с ферромагнитным покрытием. По окружности барабана размещается 512 записанных сигналов, представляющих значения одного разряда всех чисел. Таким образом, в магнитном запоминающем устройстве может одновременно храниться 512 чисел. Каждой образующей, на которой расположено число, присваивается определенный адрес. В машине для магнитной памяти используются адреса от № 0 до № 511.

В электростатическом запоминающем устройстве каждая двоичная цифра запоминаемого числа представляется в виде электрического заряда большей или меньшей величины, расположенного на определенном участке электрического экрана электронно-лучевой трубки. Количество трубок равно количеству разрядов числа (34). На экране одной трубки хранятся значения одного разряда всех запоминаемых чисел. Всего на экране трубки размещается 512 элементов, заряды на которых не зависят друг от друга. Таким образом, в электростатическом запоминающем устройстве может храниться одновременно 512 чисел или инструкций. Каждому элементу на экране присвоен определенный адрес. Для данного ЗУ использовались адреса от № 512 до № 1023. Выбор нужного элемента на экране производится изменением положения электронного луча. Электростатическое ЗУ было наиболее быстродействующим из осуществленных к 1952 году запоминающих устройств.

Внешнее запоминающее устройство, использующее магнитную ленту, построено на базе обычного магнитофона МАГ-8. Режим работы внешнего ЗУ отличается от режима работы внутренних запоминающих устройств: его нельзя использовать для оперативной работы арифметического узла.

В порядке дальнейшего усовершенствования ЭВМ М-2 было разработано, изготовлено и введено в состав машины ферритовое запоминающее устройство, работающее по принципу совпадения токов, объемом 4096 34-разрядных чисел. Возглавил эту работу М.А. Карцев. К ее выполнению подключились молодые специалисты О.В. Росницкий, Е.Н. Филинов, Л.В. Иванов, Ю.Н. Глухов, Р.П. Шидловский. Введение в состав машины М-2 ферритовой памяти значительно повысило ее эксплуатационные характеристики.

*Программный датчик* предназначен для автоматического управления выполнением операций. Программный датчик выбирает из внутреннего запоминающего устройства в нужном порядке инструкции, закодированные в виде чисел, расшифровывает их и, в соответствии с имеющимися в них указаниями, выбирает необходимые числа, производит над этими числами те или иные действия и записывает результаты этих действий в указанные ячейки внутреннего запоминающего устройства; когда это предусмотрено программой, программный датчик выводит полученные результаты на печать или на магнитную ленту; пополняет содержимое внутренних запоминающих устройств через входные устройства или с магнитных лент; производит передачу чисел из одного запоминающего устройства в другое и останавливает работу в конце вычислений. Для контроля машины в программном датчике предусмотрена возможность осуществления полуавтоматической (шаговой) работы.

### **Устройства ввода и вывода**

Входные устройства предназначены для первоначального заполнения внутренних запоминающих устройств числами и инструкциями и для пополнения содержимого запоминающих устройств в процессе работы. В машине имеется два входных устройства: фотоэлектрическое и электромеханическое. Оба устройства считывают информацию с бумажной перфоленки.

Используется обычная телеграфная бумажная лента на пять дорожек (шестая дорожка ведущая). Комбинация на пяти дорожках, относящаяся к одной цифре, прочитывается либо ошупывающим механизмом в электромеханическом устройстве ввода, либо световым лучом в фотоэлектрическом устройстве и передается в регистр ввода, состоящий из пяти триггеров.

Скорости работ устройств различны: электромеханическое устройство передает в машину одно число за 2 секунды, фотоэлектрическое – за 0,03 секунды. В машине имеется также вспомогательное клавишное устройство ввода, при помощи которого с пульта можно ввести в машину цифру за цифрой любое число или инструкцию.

Выходное устройство предназначено для печатания окончательных результатов вычислений. В качестве выходного устройства используется рулонный телеграфный аппарат – телетайп. Выходные данные печатаются в виде таблиц 8-значных чисел в десятичной или шестнадцатеричной системах и могут одновременно наноситься на перфоленку. Скорость печатания – одно число за 2,5 секунды.

### **Элементная база**

Основные электронные схемы машины М-2 (триггеры, клапаны, усилители и др.) строились на радиолампах, логические схемы – на полупроводниковых приборах (малогобаритных купроксных выпрямителях КВМП-2-7). Использовались радиолампы 6Н8С, 6Ж4Б, 6П9. Всего в машине 1879 электронных ламп.

## Конструкция

Основные узлы и блоки машины смонтированы в четырех шкафах: в шкафу № 1 – арифметический узел, в шкафу № 2 – программный датчик, в шкафу № 3 – электроника магнитного ЗУ и устройств ввода-вывода, в шкафу № 4 – электростатическое ЗУ (позднее вместо этого шкафа был установлен шкаф с ферритовым ЗУ). Все шкафы были установлены на одном постаменте. Шкаф № 5 (питание) был вмонтирован в постамент.

К постаменту примыкали пульт управления и два стола, скрепленные с постаментом: справа от пульта – стол с телетайпом, слева – с входными устройствами и магнитофоном.

Вся электронная часть машины собрана на съемных субблоках. Каждый узел машины состоял из блоков. Монтаж блоков выполнялся на шасси, прикрепленных к раме шкафа. На шасси блока располагались гнезда штепсельных разъемов, с помощью которых осуществлялось электрическое соединение съемных субблоков с монтажом данного узла.

Соединительный монтаж между субблоками на шасси блоков и между блоками внутри каждого шкафа выполнялся либо в виде тонких шин, либо в виде жгутовых перемычек. Соединения между шкафами и пультом осуществлялись кабелями, оканчивающимися стандартными разъемами.



*Конструкции субблоков в ЭВМ М-2*

## Эксплуатация машины

С вводом машины М-2 в эксплуатацию летом 1953 года началась ее постоянная работа. Для пользователей было разработано математическое обеспечение машины, включающее библиотеку обслуживающих программ (ввод-вывод, служебные программы, программы элементарных функций и др.). Была создана группа программистов во главе с доктором физико-математических наук А.Л. Брудно, которая консультировала пользователей при решении задач сторонними организациями. Постоянно действовал семинар программистов. Машина работала круглосуточно. Воскресное время распоряжением И.С. Брука было отдано программистам для теоретической разработки интеллектуальных, игровых, узнающих и других программ. По мере эксплуатации машины развивалось и математическое обеспечение. В частности, А.Л. Брудно предложил и реализовал метод содержательных обозначений (прообраз АВТОКОДА, АССЕМБЛЕРА).

На машине М-2 решали свои задачи многие научные институты и конструкторские организации: Институт атомной энергии, Институт теоретической и экспериментальной физики, Физический институт АН СССР, Центральный институт прогнозов, МАИ, Институт нефти и газа им. Губкина, МГУ, проектная организация «Стальпроект» и др.

Решались многие научные и инженерно-проектные задачи. Вот только некоторые из них:

- Расчет деполяризации спинов пучка электронов при прохождении через вещество.
- Расчет флюктуаций при ионизации тонких пластинок.
- Расчет энергетических уровней для эллиптических моделей ядра.
- Расчеты по ядерным реакторам различных типов (до 13 уравнений в системе).
- Расчет по большим ускорителям и подбор формы полюсов магнитов.
- Расчет опор Братской ГЭС.
- Размещение скважин на Ставропольском месторождении.
- Регулирование напряжения в дальних линиях электропередач.

На промышленной выставке ВДНХ в 1956 году был выставлен выносной пульт ЭВМ М-2, связанный телефонной линией с машиной, расположенной на Ленинском проспекте. Машина решала задачи, задаваемые с этого пульта, и выдавала распечатки результатов решения на рулонный телетайп, размещенный рядом с выносным пультом на ВДНХ. Когда начинался сеанс работы с машиной, посетители павильона ВДНХ, где демонстрировались достижения Академии наук СССР, собирались со всего зала и с интересом наблюдали за результатами распечатки. Это был первый опыт дистанционного управления работой ЭВМ.

Свыше 15 лет вычислительная машина М-2 исправно несла свою службу.

### **Библиографический список**

1. Быстродействующая вычислительная машина М-2 / Под ред. чл.-корр. АН СССР И.С. Брука. М., 1957.
2. Карцев М.А. Арифметические устройства электронных цифровых машин. М., 1958.

### **Отзывы пользователей**

...В институте построили отличную трехадресную машину М-2. Строил М-2 талантливый инженер М.А. Карцев. Он придумал очень удобную систему команд. Вероятно, система команд чем-то похожа на стихотворение – ее сочинение не терпит соавторства. Может быть, именно поэтому система команд М-2 получилась такой цельной – не сборная солянка, а поэма в кодах.

По нынешним временам машина М-2 показалась бы очень убогой – всего 512 ячеек электронной и 512 барабанной памяти (со сквозной нумерацией). Но в 1954 году такая машина считалась отличной: у нее была плавающая запятая, а скорость доходила до 2000 действий в секунду.

*Кронрод А.С. Беседы о программировании. М., 2001.*

...Работая примерно с такой же скоростью, как и ЭВМ «Стрела», машина М-2 содержала в 4 раза меньше электронных ламп, потребляла в 7 – 8 раз меньше электроэнергии, занимала в 10 раз меньшую площадь (всего 22 кв. метра) и стоила примерно в 10 раз дешевле. Эти достижения были обусловлены применением обычных осциллографических ЭЛТ в качестве элементов ЗУ и полупроводниковых диодов в логических схемах...

*Филинов Е.Н. История информатики в России (бывшем СССР) // Труды Международной конференции-выставки «Открытые системы – путь к новому миру». 17 – 20 апреля 1995. Москва.*

## ЭВМ М-3

**Ю.В. Рогачев**

НИИ вычислительных комплексов им. М.А. Карцева  
Москва, РФ

Универсальная малогабаритная цифровая электронно-вычислительная машина М-3 по своим эксплуатационным характеристикам предназначалась для использования в условиях научно-исследовательских институтов и конструкторских бюро. К числу задач, решаемых на машине М-3, относятся интегрирование обыкновенных дифференциальных уравнений и уравнений в частных производных (как линейных, так и нелинейных), решение систем линейных уравнений со многими неизвестными, решение алгебраических и трансцендентных уравнений и т.п.

Научный руководитель разработки член-корреспондент АН СССР И.С. Брук.

Технический проект ЭВМ М3 был разработан в инициативном порядке в 1953 году сотрудниками лаборатории электросистем Энергетического института АН СССР, преобразованной в 1956 году в Лабораторию управляющих машин и систем (ЛУМС) АН СССР.

Основные разработчики технического проекта: Н.Я. Матюхин (руководитель разработки), В.В. Бельнский, Ю.Б. Пржиемский, Н.А. Дорохова, А.Б. Залкинд, Г.И. Танетов, А.Н. Патрикеев, А.П. Морозов.

Правительственного задания на создание вычислительной машины М-3 не было, поэтому не был решен и вопрос ее изготовления. Судьбу машины решила заинтересованность трех академиков – А.Г. Иосифьяна (ВНИИЭМ), В.А. Амбарцумяна (Академия наук Армении) и С.П. Королева, которые договорились с И.С. Бруком о совместном завершении работ и изготовлении трех образцов машины на производственной базе ВНИИЭМ.

Разработчики ЛУМС АН СССР совместно с сотрудниками ВНИИЭМ и других заинтересованных организаций доработали технический проект, завершив доработку выпуском конструкторской документации для промышленного производства машины. В дополнение к разработчикам от ЛУМС к этой работе присоединились В.М. Долкерт, В.М. Каган, Т.П. Лопато, В.Н. Овчаренко, А.Я. Яковлев, Б.Б. Мелик-Шахназаров, А.П. Толмасов, В.А. Морозов, И.А. Скрипкин, А.В. Пипинов, В.Н. Семенова.

Организацию работ во ВНИИЭМе по доработке конструкторской и технологической документации, подготовке производства, разработке программного обеспечения, подготовке и проведению Государственных испытаний осуществлял к.т.н. Б.М. Каган.

В 1956 году опытным производством ВНИИЭМ были изготовлены и с участием разработчиков отлажены три комплекта ЭВМ М-3. Первый головной образец был оставлен во ВНИИЭМе для подготовки и проведения государственных испытаний, второй комплект получил Ереванский математический институт Академии наук Армении, третий комплект – организация С.П. Королева.

К этому времени было создано и необходимое программное обеспечение. Таким образом, на государственные испытания ЭВМ М-3 предъявлялась с уже отработанным программным обеспечением. Государственные испытания успешно завершились в конце 1956 года. По результатам этих испытания вычислительная машина М-3 была рекомендована для серийного производства. Но планами существующих заводов-изготовителей производство машины М-3 не предусматривалось. Помогло случайное совпадение по срокам: именно в это время завершалось строительство Минского завода вычислительной техники, а загрузить его было нечем. Тогда Госпланом СССР было принято решение о передаче ЭВМ М-3 в серийное производство на этот новый завод. Так с 1958 года в Минске началось серийное производство машины М-3, которое продолжалось несколько лет. Затем конструкторская документация на заводе была переработана и машина стала выпускаться с названием «Минск».

По конструкторской документации ЭВМ М-3 и с технической помощью ВНИИЭМ в Армении была построена ЭВМ «Раздан», первая ЭВМ в организованном в 1956 году Ереванском НИИ математического машиностроения. Таким образом, создание ЭВМ М-3 сыграло важную роль в становлении разработок и производства электронной вычислительной техники в Белоруссии и Армении. Значительный вклад в это внесли Г.П. Лопато (Минский завод) и Г.Г. Мелик-Пашаев (Ереванский НИИ математического машиностроения), получившие опыт в работе над ЭВМ М-3 во ВНИИЭМе.

### **Основные технические характеристики**

|                                    |                                                       |
|------------------------------------|-------------------------------------------------------|
| Система представления чисел.....   | с фиксированной запятой                               |
| Количество двоичных разрядов ..... | 30, кроме разряда знака,                              |
| .....                              | что соответствует 9 разрядам десятичных чисел         |
| Арифметический узел .....          | параллельного типа                                    |
| Внутреннее ЗУ .....                | на магнитном барабане с параллельным выбором          |
| .....                              | емкостью 2048 чисел*                                  |
| Средняя скорость работы .....      | 30 операций в секунду**                               |
| Система кодирования .....          | двухадресная                                          |
| Скорость вывода информации .....   | 30 чисел в минуту транзиттером Т-50,                  |
| .....                              | 1200 чисел в минуту фотоэлектрическим транзиттером.   |
| .....                              | 30 чисел в минуту (результаты печатаются в десятичной |
| .....                              | или в восьмеричной системах)                          |

\* Объем запоминающего устройства может быть увеличен до 4096 чисел после подключения второго шкафа ЗУ.

\*\* Предусмотрена возможность подключения к машине отдельного шкафа быстродействующего ЗУ, что обеспечивает увеличение скорости работы в среднем до 1500–2000 операций в секунду.

### **Основные узлы машины**

В состав машины М-3 входят четыре основных узла: арифметический узел (АУ), программный датчик (ПД), запоминающее устройство (ЗУ) и устройства ввода и вывода.

*Арифметический узел.* АУ параллельного типа. Это позволяет получить более высокие скорости выполнения арифметических операций. Время выполнения основных арифметических операций: сложение – 60 мксек, вычитание – 75–120 мксек, умножение – 1900 мксек, деление – 2000 мксек.

Арифметический узел состоит из четырех триггерных регистров А, В, С и Д. Регистр Д выполняет функции параллельного 30-разрядного сумматора и переносит единицы перед образованием суммы. По своим функциям он существенно отличается от остальных регистров, в нем происходит пробег единицы переноса перед сложением и поразрядное суммирование. Регистр С является связующим звеном АУ с другими узлами машины. Через регистр С передается информация в ЗУ при вводе, передаются коды инструкций программы из ЗУ в ПД, осуществляется перенос кодов из одной ячейки ЗУ в другую его ячейку и обеспечивается вывод информации из ЗУ к устройствам печати.

До начала арифметических операций числа в регистрах располагаются следующим образом: в регистре А – слагаемое, вычитаемое, множимое, делитель; в регистре В – слагаемое, уменьшаемое, делимое; в регистре С – множитель. Исходные числа при логическом умножении размещаются в регистрах А и С.

По окончании операции сумма, разность, произведение образуются в регистре В и передаются в регистр С. Частное и результат логического умножения образуются в регистре С и передаются в регистр В. Таким образом, во всех случаях по окончании действия результат находится в регистрах В и С одновременно, где и сохраняется до следующего приема в них. Это позволяет использовать результат, сохраненный в регистре В, при последующих операциях.

Устройство регистра А позволяет производить следующие элементарные операции:

- параллельный прием информации из регистра С,
- взятие обратного кода содержащейся в регистре информации,
- гашение (установка на «0» всех триггеров) регистра.

В регистре В можно осуществить:

- гашение регистра,
- взятие обратного кода,
- параллельный прием из регистра С,
- параллельную передачу в регистр С,
- сдвиг содержимого вправо и влево.

В регистре С можно осуществить:

- гашение регистра,
- параллельный прием из ЗУ,
- параллельный прием из регистра В,
- параллельную передачу в регистры А и В и в регистры СР и БО программного датчика,
- последовательный ввод их в УВВ и последовательно-параллельный вывод кодов из АУ в УВВ,
- сдвиг содержимого вправо и влево,
- образование результата логического умножения,
- ручной набор информации в регистр.

*Программный датчик.* ПД автоматически управляет узлами машины АУ, ЗУ, УВВ. В процессе решения задачи программный датчик в соответствии с программой извлекает числа из ЗУ, размещает их в регистрах АУ, выполняет над ними заданную операцию, направляет результат в нужную ячейку ЗУ и выводит его (печатает), если это задано инструкцией. В процессе выполнения этого цикла, соответствующего выполнению одной инструкции программы, определяется адрес инструкции, которая должна быть выполнена в следующем цикле. Все эти действия, соответствующие одной инструкции, программный датчик выполняет в определенной последовательности в течение восьми тактов (0–7), причем содержание тактов изменяется в зависимости от выполняемой инструкции.

Программный датчик состоит из следующих блоков:

- местного программного датчика (МПД),
- блока операций (БО),
- распределителя импульсов (РИ),
- селекционного регистра (СР),
- пускового регистра (ПР).

Блок МПД предназначен для управления арифметическим узлом при выполнении им следующих операций: сложение, вычитание, умножение, деление, ввод или вывод группами по 3 или 4 разряда, логическое умножение. В блоке МПД помещены также три триггера знака регистров А, В, и С.

БО предназначен для хранения и расшифровки кода операций очередной инструкции программы, выполняемой машиной. Код операции расположен в 1–6 разрядах регистра С. Эти разряды связаны соответственно с 1–6 триггерами регистра блока операций. В соответствии с принятой системой кодов блок операции выдает в блоки МПД и в узел ввода-вывода команды, вызывающие выполнение операции, заданной инструкцией. Кроме того, блок выдает в ЗУ команду «запись» и в том случае, если

инструкция предусматривала запись результатов операции. В противном случае по окончании операции БО посылает в блок распределения импульсов (РИ) команду «запуск» РИ, вызывающую начало следующего цикла работы.

Блок РИ. Машина М-3 работает асинхронно, т.е. в ней нет единых для всего устройства тактирующих импульсов, которые и определяли бы скорость работы. Каждый из узлов машины получает импульс-команду, выполняет заданные операции и посылает импульс-ответ, означающий, что операция выполнена и можно приступить к выполнению следующей операции. Получив ответ об окончании операции, блок РИ посылает в ЗУ команду чтения следующей инструкции и одновременно передает из ПР в СР адрес этой инструкции. После того как из ЗУ придет импульс ответа об окончании чтения инструкции, в зависимости от кода операции РИ посылает в блоки ЗУ, БО, СР, ПР, АУ и МПД различные управляющие импульсы. Весь цикл работы РИ состоит из восьми тактов (0–7), число пройденных тактов отсчитывается счетчиком РИ. Содержание тактов может изменяться в зависимости от кода выполняемой инструкции.

СР и ПР. 12-разрядные триггерные регистры ПР и СР предназначены для хранения и преобразования адресов чисел и команд, выбираемых из ЗУ. Каждый триггер СР связан через дешифраторы несоответствия с триггером счетчика маркерных импульсов ЗУ. Совпадение состояния регистра СР и текущего адреса в счетчике фиксируется схемой, выполняется инструкция в зависимости от поданной в ЗУ команды. В регистре ПР в процессе выполнения данной инструкции образуется адрес следующей инструкции программы. Обычно инструкции следуют одна за другой в порядке их записи в ЗУ, поэтому в регистре ПР режим счетчика. Но в случае выполнения инструкций условного и безусловного переходов, когда адрес следующей инструкции не является следующим по порядку номеров, информация в ПР поступает из СР. Все импульсы, управляющие работой СР и ПР, поступают из блока РИ. Кроме этих основных функций регистры СР и ПР позволяют осуществлять остановку на нужном адресе и электронный вывод информации.

*Запоминающее устройство.* В качестве запоминающего устройства машины М-3 применяется вращающийся барабан, покрытый ферромагнитным слоем, с системой головок для записи и чтения сигнала. Это магнитное запоминающее устройство предназначено для хранения инструкций программы, исходных данных вспомогательных чисел, промежуточных и конечных результатов, получаемых в процессе решения задачи. Запоминающее устройство параллельного типа. Запись и чтение производятся одновременно по 31 каналу (30 разрядов числа и 1 разряд знака) на одной из образующих барабана, местонахождение которой определяется порядковым номером маркерного импульса. Для получения последовательности маркерных импульсов на поверхности барабана по окружности записано 2048 импульсов. Против дорожки, где произведена эта запись, помещается маркерная головка, сигналы которой и формируют маркерные импульсы. Другая головка – нулевая – расположена так, что в слое против нее записан только один сигнал, формирующий импульс нуля, соответствующий по времени промежутку между 2048-м и 1-м маркерными импульсами.

#### **Технические данные запоминающего устройства**

|                                  |                                      |
|----------------------------------|--------------------------------------|
| Емкость.....                     | 2048 30-разрядных двоичных чисел     |
| Среднее время выбора.....        | 10 мсек                              |
| Плотность записи .....           | 3,2 имп./мм                          |
| Число оборотов.....              | 3000 в минуту                        |
| Диаметр барабана.....            | 216 мм                               |
| Ферромагнитный слой.....         | сплав никель-кобальт толщиной 3–5 мк |
| Частота маркерных импульсов..... | 108 кгц                              |

### *Устройство ввода и вывода данных*

Устройство ввода. В машине М-3 ввод информации с бумажной перфоленты осуществляется с помощью трансмиттера Т-50. Информация, вводимая в машину, состоит из программы – последовательности инструкций, исходных цифровых данных и вспомогательных чисел. Коды инструкций всегда представляются в восьмеричной системе, а коды исходных данных и вспомогательных чисел могут быть представлены и в десятичной системе. Коды набивают на ленту на специальном перфораторе в бинарно-кодированном виде. Кроме кодов на ленте перфорируются «служебные знаки», управляющие работой устройства ввода. Устройство ввода считывает информацию с ленты, расшифровывает ее, представляет в двоичной системе и передает в запоминающее устройство. Ввод информации может быть автоматическим и ручным. Имеется возможность использовать для ввода информации фотосчитывающее устройство.

Устройство вывода. Вывод требуемых промежуточных и окончательных результатов производится на печатающее устройство – телетайп, который представляет собой электрическую печатающую машинку, имеющую ряд дополнительных механизмов: распределительный, дешифраторный, перфорационный и реле. Все эти механизмы во взаимодействии преобразуют комбинацию электрических сигналов, подаваемых на вход распределительного механизма, в удар одного из рычагов механизма печати. Наряду с печатью перфорационный механизм может перфорировать печатающие знаки на ленте. В машине М-3 используется рулонный телетайп, снабженный реперфоратором, благодаря чему можно получать результаты вычислений на перфорированной ленте. Ленту можно использовать для долговременного хранения и последующего непосредственного ввода информации в машину. Управление печатью результатов с нужными интервалами и с определенным количеством столбцов осуществляется при помощи стандартных шаговых искателей ШИ-25/8. Имеется возможность электронного вывода информации. Электронный вывод предназначен для воспроизведения на экране осциллографа графика, на котором по оси ординат откладывается величина, пропорциональная пяти старшим разрядам выводимого числа, а по оси абсцисс – величина, пропорциональная числу прошедших выводов.

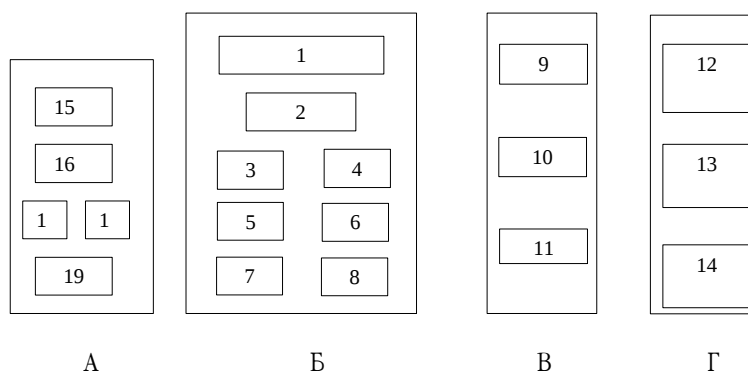
### **Электрическое питание машины М-3**

Устройство питания состоит из электромашинного агрегата и шкафа питания. В электромашинный агрегат входят асинхронный двигатель А-62, генератор ЧС-7 повышенной частоты 240 в, 200 гц и генератор ПН-68 постоянного тока, дающий постоянное напряжение 240 в. Агрегат питания расположен в отдельном помещении. В шкафу питания напряжение 240 в, 200 гц преобразуется в постоянные напряжения, необходимые для питания машины, производится стабилизация нескольких напряжений и распределение напряжений по шкафам машины. В шкафу питания размещены выпрямители, стабилизаторы, схемы управления, защиты и сигнализации, приборы контроля. Для питания электронных схем машины используются следующие уровни напряжений (вольт): +70, +120, +150, +240, +350, +450, -170.

Все напряжения подаются в шкафы машины с помощью гибких кабелей.

### **Конструкция машины М-3**

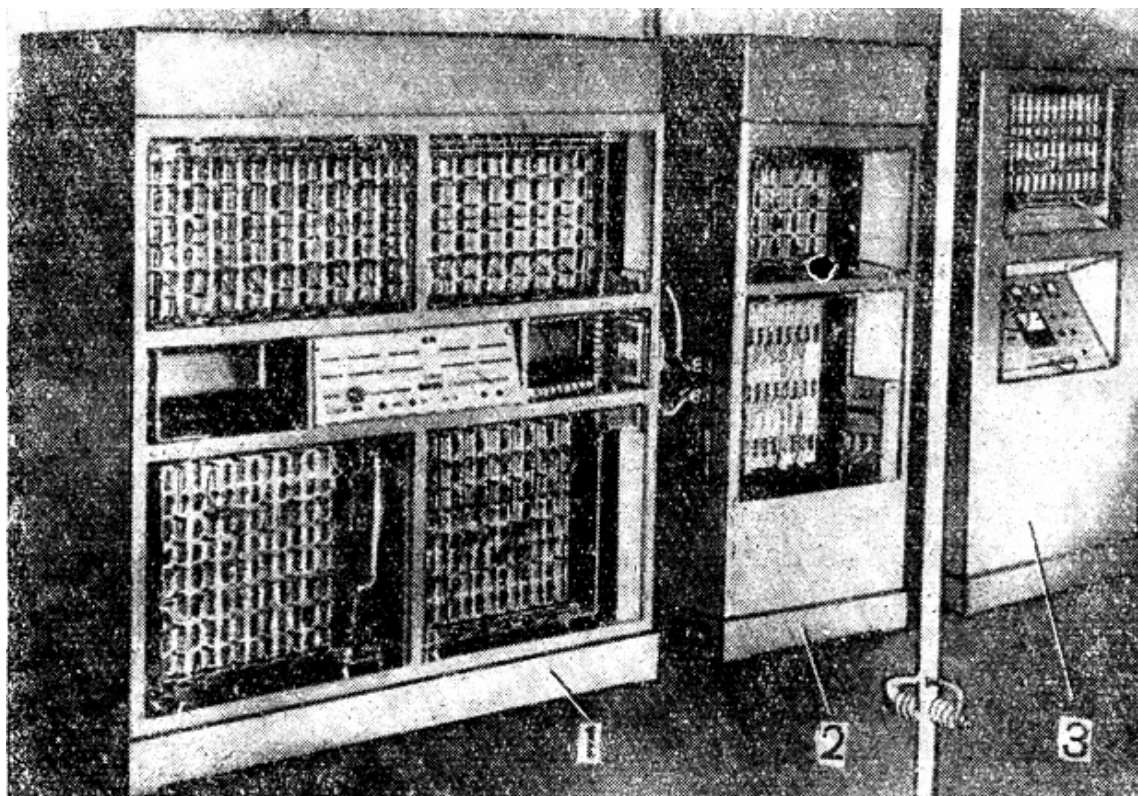
Все оборудование машины размещено в трех шкафах и на одном передвижном столике. В главном шкафу № 1 размещены арифметический узел, программный датчик, электронный блок управления вводом и выводом и главный пульт. В шкафу № 2 расположено запоминающее устройство: магнитный барабан, усилители, схема управления запоминающим устройством. В шкафу № 3 размещено оборудование устройства питания: выпрямители, стабилизаторы, системы включения и выключения и защиты машины. На подвижном столике установлены телетайп, пульт устройства ввода и вывода, трансмиттер, устройство фотоввода, блоки реле и шаговых искателей.



**Схема расположения узлов и блоков машины М-3:**

*А* – подвижный столик, *Б* – шкаф № 1, *В* – шкаф № 2, *Г* – шкаф № 3;

*1* – арифметический узел ; *2* – главный пульт управления; *3* – блок операций ; *4* – местный программный датчик;  
*5* – электронный блок устройства ввода и вывода; *6* – селекционный регистр ; *7* – пусковой регистр; *8* – блок распределения импульсов; *9* – счетчик маркерных импульсов, схема сравнения и блок управления магнитной памятью;  
*10* – усилители записи и чтения; *11* – магнитный барабан; *12* – стабилизаторы; *13* – пульта управления шкафом питания; *14* – выпрямители; *15* – телетайп; *16* – пульт устройства ввода и вывода; *17* – трансмиттер;  
*18* – устройство фотоввода; *19* – блоки реле и шаговых искателей



**Внешний вид машины М-3:**

*1* – главный шкаф; *2* – шкаф запоминающего устройства; *3* – шкаф питания

Узлы машины смонтированы на дюралевых рамах, укрепленных на каркасе шкафов. В шкафу № 1 установлены три рамы, в шкафу № 2 – две рамы. Поле каждой рамы представляет решетку, на которой укрепляются стандартные посадочные площадки для сменных субблоков. Сменные субблоки имеют три основных габаритных размера: двухламповый – 90 x 48 x 120 мм, одноламповый – 60 x 48 x 120 мм и четырехламповый – 208 x 48 x 120 мм. Субблок соединяется с площадкой при помощи разъемов (20- или 14-контактного).

Для охлаждения ламп и отвода тепла во всех шкафах введена принудительная вентиляция. Система вентиляции – разомкнутая, воздух засасывается с нижней части каждой рамы через решетчатое окошко, проходит через ряды ламп и выбрасывается вентиляторами наружу.

Площадь, занимаемая отдельными узлами машины:

- шкаф № 1 (арифметический узел и устройство управления) ..... 1,1 м<sup>2</sup>
- шкаф № 2 (запоминающее устройство) ..... 0,7 м<sup>2</sup>
- шкаф № 3 (электрическое питание) ..... 0,9 м<sup>2</sup>
- подвижный столик (устройства ввода и вывода) ..... 0,6 м<sup>2</sup>

### Программирование задач

Система кодирования инструкций в М-3 – двухадресная. Код инструкции занимает все 30 разрядов чисел, кроме 31-го разряда знака. Инструкция может иметь произвольный знак. Шесть старших разрядов инструкции отводятся под код операции, остающиеся 24 разряда – под код адресов, на каждый адрес по 12 разрядов. Инструкции удобно кодировать в восьмеричной системе, объединяя каждые три двоичных разряда в один восьмеричный. Таким образом, коды операций имеют по два восьмеричных разряда, адреса – по четыре. Инструкции должны перфорироваться на ленте перфоратора и вводиться в машину также в восьмеричной системе. Ниже приводится структура кодов чисел и инструкций.

#### Структура кодов чисел и инструкций

|                |   |   |   |   |   |                      |   |   |    |    |    |    |    |    |    |    |    |                      |    |    |    |    |    |    |    |    |    |    |    |  |  |  |
|----------------|---|---|---|---|---|----------------------|---|---|----|----|----|----|----|----|----|----|----|----------------------|----|----|----|----|----|----|----|----|----|----|----|--|--|--|
| 1              | 2 | 3 | 4 | 5 | 6 | 7                    | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19                   | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |  |  |  |
| 1              |   | 2 |   |   |   | 3                    |   | 4 |    |    |    | 5  |    |    |    | 6  |    |                      |    | 7  |    |    | 8  |    |    | 9  |    |    | 10 |  |  |  |
| Код операции   |   |   |   |   |   | Восьмеричные разряды |   |   |    |    |    |    |    |    |    |    |    | Восьмеричные разряды |    |    |    |    |    |    |    |    |    |    |    |  |  |  |
|                |   |   |   |   |   | Первый адрес         |   |   |    |    |    |    |    |    |    |    |    | Второй адрес         |    |    |    |    |    |    |    |    |    |    |    |  |  |  |
| Код инструкции |   |   |   |   |   |                      |   |   |    |    |    |    |    |    |    |    |    |                      |    |    |    |    |    |    |    |    |    |    |    |  |  |  |

Коды команд арифметических операций и логического умножения образуются следующим образом: вторая цифра кода означает вид операции (сложение, вычитание и др.), первая цифра указывает признак (необходимо ли произвести запись результата в запоминающее устройство и т.д.). Принятые значения кодирующих цифр следующие:

Первая цифра кода – вспомогательный признак:

- 0\* – операция над числами из первого и второго адресов. Результат записывается в ЗУ по второму адресу;
- 1\* – операция над числами по первому и второму адресу. Результат не записывается;
- 2\* – операция над числом из первого адреса и результатом предыдущей операции, хранящимся в арифметическом узле. Результат записывается в ЗУ по второму адресу;
- 3\* – операция над числом из первого адреса и результатом предыдущей операции, хранящейся в арифметическом узле. Результат в ЗУ не записывается;
- 4\* – то же, что и для 0, но результат, помимо записи в ЗУ, печатается;

5\* – то же, что и для 1, но действия производятся над модулями чисел;

6\* – то же, что и для 2, но результат помимо записи в ЗУ, печатается;

7\* – то же, что и для 3, но действия производятся над модулями чисел.

Вторая цифра кода – основной признак:

\*0 – сложение; \*1 – вычитание; \*2 – деление; \*3 – умножение; \*6 – логическое умножение.

### Код команд машины М-3

| Код | Условное обозначение | Исполняемое действие                                                                                                                 |
|-----|----------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 00  | +                    | К содержимому первого адреса прибавляется содержимое второго адреса и записывается в ЗУ по второму адресу                            |
| 10  | +.                   | К содержимому первого адреса прибавляется содержимое второго адреса, но в ЗУ не записывается                                         |
| 20  | ↓+                   | К результату предыдущего действия прибавляется содержимое первого адреса. Результат записывается по второму адресу                   |
| 30  | ↓+.                  | К результату предыдущего действия прибавляется содержимое первого адреса, но в ЗУ не записывается                                    |
| 40  | +Π                   | К содержимому первого адреса прибавляется содержимое второго адреса. Результат записывается в ЗУ по второму адресу и печатается      |
| 50  | i + l,               | К модулю содержимого второго адреса прибавляется модуль содержимого первого адреса, но в ЗУ не записывается                          |
| 60  | ↓+Π                  | К результату предыдущего действия прибавляется содержимое первого адреса. Результат записывается в ЗУ по второму адресу и печатается |
| 70  | ↓/+/                 | К модулю результата предыдущего действия прибавляется модуль содержимого первого адреса, но в ЗУ не записывается                     |
| 01  | -                    | Из содержимого второго адреса вычитается содержимое первого адреса. Результат записывается в ЗУ по второму адресу                    |
| 11  | -,                   | Из содержимого второго адреса вычитается содержимое первого адреса, но результат в ЗУ не записывается – остается в АУ                |
| 21  | ↓-                   | Из результата предыдущего действия вычитается содержимое первого адреса. Результат записывается в ЗУ по второму адресу.              |
| 31  | ↓-,                  | Из результата предыдущего действия вычитается содержимое первого адреса. Результат не записывается                                   |
| 41  | -Π                   | Из содержимого второго адреса вычитается содержимое первого адреса. Результат записывается по второму адресу и печатается            |
| 51  | /-/,                 | Из модуля содержимого второго адреса вычитается модуль содержимого первого адреса. Результат не записывается                         |
| 61  | ↓-Π                  | Из результата предыдущего действия вычитается содержимое первого адреса. Результат записывается по второму адресу и печатается       |
| 71  | ↓/-/                 | Из модуля результата предыдущего действия вычитается модуль содержимого первого адреса, но в ЗУ не записывается                      |
| 02  | :                    | Содержимое второго адреса делится на содержимое первого адреса. Результат записывается в ЗУ по второму адресу.                       |
| 12  | : ,                  | Содержимое второго адреса делится на содержимое первого адреса. Результат в ЗУ не записывается                                       |
| 22  | ↓:                   | Результат предыдущего действия делится на содержимое первого адреса. Результат записывается в ЗУ по второму адресу                   |
| 32  | ↓: ,                 | Результат предыдущего действия делится на содержимое первого адреса. Результат в ЗУ не записывается                                  |
| 42  | :Π                   | Содержимое второго адреса делится на содержимое первого адреса. Результат записывается в ЗУ по второму адресу и печатается           |
| 52  | / : /,               | Модуль содержимого второго адреса делится на модуль содержимого первого адреса, но в ЗУ не записывается                              |
| 62  | ↓:Π                  | Результат предыдущего действия делится на содержимое первого адреса. Результат записывается в ЗУ по второму адресу и печатается.     |
| 72  | ↓/ : /               | Модуль результата предыдущего действия делится на модуль содержимого первого адреса, но в ЗУ не записывается                         |

| Код                                          | Условное обозначение | Исполняемое действие                                                                                                                                                                         |
|----------------------------------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 03                                           | X                    | Содержимое второго адреса умножается на содержимое первого адреса. Результат записывается в ЗУ по второму адресу                                                                             |
| 13                                           | X ,                  | Содержимое второго адреса умножается на содержимое первого адреса. Результат в ЗУ не записывается                                                                                            |
| 23                                           | ↓X                   | Результат предыдущего действия умножается на содержимое первого адреса. Результат записывается в ЗУ по второму адресу                                                                        |
| 33                                           | ↓X ,                 | Результат предыдущего действия умножается на содержимое первого адреса. Результат в ЗУ не записывается                                                                                       |
| 43                                           | XП                   | Содержимое второго адреса умножается на содержимое второго адреса. Результат записывается в ЗУ по второму адресу и печатается                                                                |
| 53                                           | / X /,               | Модуль содержимого второго адреса умножается на модуль содержимого первого адреса, но в ЗУ не записывается                                                                                   |
| 63                                           | ↓XП                  | Результат предыдущего действия умножается на содержимое первого адреса. Результат записывается в ЗУ по второму адресу и печатается                                                           |
| 73                                           | ↓/X/,                | Модуль результата предыдущего действия умножается на модуль содержимого первого адреса, но в ЗУ не записывается                                                                              |
| 06                                           | Λ                    | Содержимое второго адреса логически умножается на содержимое первого адреса. Результат записывается в ЗУ по второму адресу                                                                   |
| 16                                           | Λ ,                  | Содержимое второго адреса логически умножается на содержимое первого адреса. Результат остается в АУ                                                                                         |
| 26                                           | ↓Λ                   | Результат предыдущего действия логически умножается на содержимое первого адреса. Результат записывается в ЗУ по второму адресу                                                              |
| 36                                           | ↓Λ ,                 | Результат предыдущего действия логически умножается на содержимое первого адреса. Результат в ЗУ не записывается                                                                             |
| 46                                           | Λ П                  | Содержимое второго адреса логически умножается на содержимое первого адреса. Результат записывается в ЗУ по второму адресу и печатается                                                      |
| 56                                           | / Λ /,               | Модуль содержимого второго адреса логически умножается на модуль содержимого первого адреса. Результат в ЗУ не записывается                                                                  |
| 66                                           | Λ П                  | Результат предыдущего действия логически умножается на содержимое первого адреса. Результат записывается в ЗУ по второму адресу и печатается                                                 |
| 76                                           | ↓/ Λ /,              | Модуль результата предыдущего действия логически умножается на содержимое первого адреса, но в ЗУ не записывается.                                                                           |
| 07; 27                                       | Вв                   | Ввод чисел. Число с перфорированной ленты записывается по второму адресу, На регистре не сохраняется                                                                                         |
| 05; 15                                       | Пч                   | Содержимое первого адреса передается во второй регистр и сохраняется в АУ на регистре второго числа                                                                                          |
| 45; 55                                       | ПчП                  | Содержимое первого адреса передается во второй адрес и печатается, но в АУ не сохраняется                                                                                                    |
| 24                                           | ПУ                   | Передача управления. Следующая инструкция берется по первому адресу. Результат предыдущего действия записывается по второму адресу и сохраняется в АУ                                        |
| 64                                           | Пуп                  | То же, что и операция 24, но число еще и печатается                                                                                                                                          |
| 74                                           | ПУ                   | Передача управления. Следующая инструкция берется по второму адресу. В АУ остается модуль результата предыдущего действия                                                                    |
| 34                                           | УП                   | Условный переход. Следующая инструкция берется по второму адресу, если результат предыдущего действия имел знак «+», и по первому адресу, если результат предыдущего действия имел знак «-». |
| 04<br>14<br>44<br>54<br>17<br>37<br>57<br>77 | Останов              | Операции останова отличаются друг от друга содержанием регистров арифметического узла, а также селекционного и пускового регистров.                                                          |

### Библиографический список

1. Бельнский В.В., Долкарт В.М., Каган Б.М., Лопато Г.П., Матюхин Н.Я. Малогабаритная электронная вычислительная машина М-3. М., 1957.
2. Каган Б.М. История из первых рук // PC WEEK | RE. 2003. № 46.

## АЛГОРИТМИЧЕСКАЯ ЭВМ

**З.Д. Алексеева, Я.А. Хетагуров**

Концерн «Моринформсистема – Агат», МИФИ

Москва, РФ

*chief@pcweek.ru*

В 1987 году НПО «Агат» были проведены межведомственные испытания опытного образца разработанной алгоритмической ЭВМ с непосредственной реализацией языка высокого уровня типа Фортран.

Основой являлся язык Фортран-4, который был расширен включением некоторых операторов языка PL-1 и специальных операторов и описаний, позволяющих работать в системах реального времени и в условиях необходимости защиты процесса решения от неисправностей. Для этого введены операторы, позволяющие организовывать защиту программ и файлов, параллельную работу программ, осуществлять блокировку прерываний, задавать реакции на прерывания. Языковые средства позволяют фиксировать интересующие разработчиков ситуации, имитировать возникновение сигналов прерываний, динамически, статически и по предписанию распределять память. Для защиты работы машины от последствий сбоев использовались известные аппаратурные и языковые средства, которые позволяют разбивать программы на участки, позволяющие производить повтор решения в случае сбойных ситуаций. Для обеспечения работы в реальном времени введены операторы работы со счетчиками времени и др.

Разработанный язык на базе языка FORTRAN -IV был назван RTF (real time FORTRAN).

ЭВМ имела следующие характеристики:

- числа, хранящиеся в ЗУ, имели разрядность 32 двоичных разряда;
- объем непосредственно адресуемой памяти составлял до 8192 Кбайт.

В ЭВМ могли использоваться числа следующих типов:

- короткое число с фиксированной точкой (16 разрядов);
- число с фиксированной точкой (32 разряда);
- число с фиксированной точкой двойной длины (64 разряда);
- короткая битовая строка (16 разрядов);
- битовая строка (32 разряда);
- число с плавающей точкой (32 разряда).

Все числа с плавающей и фиксированной точкой представлены в дополнительном коде.

Для представления логических переменных использовался один разряд битовой строки.

Значения TRUE -0, FALSE -1 используются для осуществления функций управления.

Для преобразования выражений предусмотрены 4 базовых регистра. Алгоритм состоит из различного вида записей. Адрес памяти образуется приформированием номера страницы, который содержится в базовом регистре, и номера полуслова, который содержится в адресной части записей видов 1, 2, 7.

Имеются следующие виды записей:

1. Нулевой вид, состоящий из 16 разрядов с тремя разрядами кода записи и 13 разрядами номера страницы памяти. Этот вид устанавливает номер страницы во втором базовом регистре.
2. Первый вид, состоящий из 16 разрядов. Адресуется простая переменная и константа (номер базового регистра, тип переменной).
3. Второй вид, состоящий из 16 разрядов. Адресуется переменная с индексом.
4. Третий вид, состоящий из 8 разрядов. Запись характеризует код операции.

5. Четвертый вид, состоящий из 16 разрядов, где 3 разряда содержат код записи, 5 разрядов – код операции, 8 разрядов – код сдвига. Запись характеризует операцию сдвига, что связано с уплотненным расположением информации в памяти и необходимостью выделять определенное количество разрядов для расшифровывания содержащейся информации.
6. Пятый вид, содержащий 8 разрядов. Этот вид записи обозначает коды стандартных функций и управляющих символов.
7. Шестой вид, состоящий из 24 до 40 разрядов, характеризует непосредственно операнд.  
Записи образуют инструкции, которые поступают в устройство управления и организуют процесс выполнения задачи.

### **Операции, выполняемые арифметическим процессором**

Арифметические операции, выполняемые в алгоритмической машине, соответствуют набору операций обычной машины, но дополнены некоторыми операциями перевода одного типа данных в другой.

### **Основные особенности устройства управления**

Выполнение инструкций в машине при таком внутреннем языке требует дополнительного объема аппаратуры. Поэтому наряду с обычными элементами устройства управления в алгоритмической машине присутствует ряд дополнительных регистров, управляющих триггеров, счетчиков. Все эти дополнительные элементы представляют собой устройство выборки и предварительной обработки инструкций.

Инструкции имеют различную длину, располагаясь в памяти без промежутков.

Для выборки инструкций предусмотрены два 32-разрядных регистра.

Наличие двух регистров вызвано необходимостью формирования в целом виде символов, входящих в состав инструкции. В качестве символов могут быть использованы следующие виды данных:

- константы 8, 16, 24, 32-разрядные;
- идентификаторы, размещаемые в ОЗУ, – 8-разрядные (переменные);
- идентификаторы, находящиеся в ДЗУ (метки, имена), – 16-разрядные;
- операции, разделители (скобки, запятые и т.д.) – 8-разрядные.

### **Структура инструкции**

Инструкция представляет собой команду, полученную после претрансляции предложения, записанного на языке высокого уровня. Претрансляция – перевод стандартно записанной информации в операторах, расположенных построчно, в содержание ячеек памяти. Длина инструкции может быть произвольной величины, кратной полубайтам. Каждая следующая инструкция начинается с полубайта, следующего за конечным полубайтом предыдущей инструкции. Исключение составляет первая инструкция программы и инструкции, на которые при выполнении программы передается управление. Их запись всегда начинается с новой ячейки, независимо от заполнения предыдущей.

Первая инструкция имеет следующую структуру:

Первый байт инструкции содержит в 2 своих старших разрядах признак начала инструкции, а в остальных – код оператора.

Второй байт инструкции включает в себя код числа символов в инструкции и признак первого символа. Значения символов следующие:

00 – идентификатор ОЗУ;

- 01 – идентификатор ДЗУ;
- 10 – операция или разделитель;
- 11 – константа.

Константа, как было сказано выше, может иметь длину от 1 до 4 байт. Для экономии памяти использовали признаки длины, хотя они усложняли анализ инструкции. При этом если в первом признаке символа был признак константы, то вместо второго признака ставился признак длины константы, затем следовал символ константы, а не признак второго символа. Если признак константы появлялся на месте второго признака, то код типа константы располагался во втором полубайте признаков на месте первого признака. В этом случае дальше прочитывался только один символ. Для символов, отличных от константы, порядок следования был общим. Значения признаков длины константы следующие:

- 00 – 1 байт;
- 01 – 2 байта;
- 10 – 3 байта;
- 11 – 4 байта.

Конец инструкции определяется как аппаратурным путем (по обнулению Сч количества символов), так и по выделению кода конца инструкции.

### Система элементов

Опытный образец ЭВМ выполнен на 133 серии.

Арифметический процессор (без схемы контроля) состоял из 412 корпусов, из них на устройство управления приходится 138 корпусов.

Время выполнения операции сложения для 1-го и 2-го типа данных составляло 0,3 мкс, для 6-го типа – от 0,7 до 9,6 мкс. Операция умножения выполнялась для 1-го и 2-го типа за 4,9 – 9,7 мкс, а для 6-го типа – за 5,2 – 10,1 мкс. Логические операции выполнялись за 0,3 мкс для 1-го, 2-го, 3-го типов и за 0,6 – 6,8 мкс для 6-го типа.

Логические операции типа «лог. НЕ» за 0,1 мкс для 4-го и 5-го типов.

### Оценка алгоритмической ЭВМ

Межведомственной комиссией по приемке машины на основании работы опытного образца при решении как вычислительных, так и системных задач были сделаны следующие выводы:

1. Программирование на языке RTL повысило производительность кодирования программы относительно кодирования на машинном языке в 5 раз;
2. Программы, написанные на языке RTL, сократились по объему по сравнению с программами на машинном языке в среднем в 3,3 раза;
3. Объем памяти, занятый программой, написанной на RTL, примерно равен объему памяти, занятому этой же программой, написанной на машинном языке, в 3–5 раз меньший, чем программа, написанная на ЯВУ и оттранслированная на машинный язык;
4. Объем, занимаемый аппаратурой, составил 120 л в бескорпусном исполнении;
5. Быстродействие ЭВМ составило около 2 млн. коротких операций в сек.

Главный конструктор Алексеева З.Д.

Научный руководитель Хетагуров Я.А.

Основные разработчики: Кузнецов Г.И., Рихтер Г.Г., Полтавец Г.Н.

## ПОДГОТОВКА СПЕЦИАЛИСТОВ ПО ИНФОРМАЦИОННЫМ И КОММУНИКАЦИОННЫМ ТЕХНОЛОГИЯМ НА БАЗЕ СЕМЕЙСТВА СТАНДАРТОВ «ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»

**Ю.А. Богоявленский**

Петрозаводский государственный университет

Петрозаводск, РФ

ybgg@cs.karelia.ru

Говорят: «Будут гурии, мед и вино –  
Все улады в раю нам вкусить суждено».  
Потому я повсюду с любимой и с чашей, –  
Ведь в итоге к тому же придем все равно.

*Омар Хайям*

### Введение

В этой статье мы будем использовать термин «Информационные и коммуникационные технологии» (ИКТ) в широком смысле как аналог принятого в США термина «Computing». Сфера ИКТ быстро расширяется, государства и частные компании инвестируют в ее развитие огромные средства. Новые парадигмы, концепции, стандарты, инструменты, прикладные системы появляются и внедряются очень быстро, дополняя и/или заменяя друг друга.

Рассматривая этот процесс, авторы [1] отмечают, что в 90-е годы XX века произошла диверсификация деятельности специалистов по ИКТ, которая привела к образованию семейства из пяти дисциплин: «ЭВМ, комплексы, системы и сети» (Computer Engineering) [2], «Информатика» (Computer Science – ИН) [3, 4], «Информационные системы» (Information Systems – ИС) [5], «Информационные технологии» (Information Technology – ИТ) [6] и «Технология разработки программного обеспечения» (Software Engineering – ТПИ) [7]. Авторы [1] полагают, что содержанием ИН по-прежнему остается теоретическая информатика, а ИС – проектирование информационных систем, генерирующих, обрабатывающих и распределяющих данные в организации с точки зрения их содержания и деловых процессов для достижения конкурентных преимуществ. Отмечено также, что ТПИ постепенно выделилась из ИН, а ИТ появилась недавно для подготовки специалистов по управлению, поддержке и планированию сложных сетевых инфраструктур, без которых не могут работать современные организации.

Естественно, что такое развитие значительно увеличивает как актуальность, так и сложность проблем разработки и внедрения образовательных стандартов для решения задачи подготовки специалистов, способных к долговременной эффективной профессиональной работе в ИКТ (Задача). Эта проблема постоянно находится в центре внимания ИКТ сообществ в США [1–7], а также ученых, учебно-методических объединений Министерства науки и образования и промышленных ассоциаций в России [9–21].

В [1] предлагается строить отдельные руководства по разработке учебных планов для подготовки специалистов в каждой из пяти дисциплин. Мы не будем рассматривать в этой статье дисциплину «ЭВМ, комплексы, системы и сети». Цель статьи – показать, что подготовку специалистов по четырем остальным дисциплинам возможно (и целесообразно) эффективно организовать в рамках семейства российских государственных образовательных стандартов «Прикладная математика и информатика», прежде всего, на базе направления подготовки бакалавров 010500 («старый» шифр 510200) с последующей (при необходимости) специализацией.

## **Мотивация использования семейства стандартов**

### **«Прикладная математика и информатика»**

Разработка стандартов подготовки для новых дисциплин является, безусловно, важной задачей. В то же время высшая школа обладает большой инерцией и скорость введения и стабилизации новых стандартов, которую она может обеспечить, значительно ниже, чем наблюдаемая сейчас скорость изменений в дисциплине ИКТ [1, 22]. С другой стороны, реагировать на диверсификацию ИКТ нужно уже сейчас. В России давно существуют стандарт по ТП – 220400 «Программное обеспечение вычислительной техники и автоматизированных систем» (инженер) и стандарт по ИС – 654700 «Информационные системы» (бакалавр, инженер, магистр). В России ожидается значительный рост потребности в специалистах по ИКТ, по некоторым прогнозам [23] – 40 000 за предстоящие пять лет. В этих условиях одной из целесообразных реакций, направленных на решение Задачи, является адаптация существующих, широко распространенных в университетах стандартов.

Важным требованием к стандартам ИКТ является гибкость, которая позволит успешно использовать их для решения Задачи в условиях диверсификации в течение достаточно длительного периода времени. Естественно, что при этом стандарт должен содержать постоянную часть, дающую основы, и переменную часть, обеспечивающую реакцию на изменения. При построении постоянной части стандарта в России естественно опираться на сложившуюся у нас традицию обеспечения фундаментальности образования. Суть ее в том, что студент изучает не набор средств решения конкретных задач, а теории, методы и подходы, лежащие в основе разработки этих средств. Фундаментальность квалификации обеспечивает высокое качество и производительность труда выпускников, их способность к самообучению и быстрой адаптации к любым нововведениям, высокий потенциал карьерного роста. Этот принцип разделяют и авторы [3, 4, раздел 9.1.6].

Роль математики в становлении и развитии ИКТ является основополагающей, поскольку профессионал ИКТ имеет дело с формальными, абстрактными концепциями и объектами. В [3, 4, раздел 9.1.1] отмечается: «Математические методы и формальные рассуждения являются составной частью большинства областей информатики. <...> Учитывая глубокую роль математики в информатике, программы обучения должны включать математические концепции как можно раньше и как можно чаще».

Более того, в моделях учебных планов по ИН для исследовательских университетов США рекомендовано включать в программу «от одного до трех и более» [3, 4, раздел 9.4.1] семестровых курсов математического анализа (заметим, что в стандарте 010500 предусмотрено до пяти семестров). В [7] рекомендовано включать в планы инженеров ТП два семестра математического анализа и отмечено, что такова практика большинства университетов Северной Америки. В перечень углубленных курсов [3, 4, раздел 9.3] включены «Комбинаторика», «Вероятность и статистика», «Методы вычислений», «Исследование операций», «Теория языков и автоматов» и другие, которые являются обязательными в стандарте 010500. В работах [1–7] также подчеркивается важность математической подготовки в пяти перечисленных в [1] (и инженерных по определению) дисциплинах ИКТ. Отметим, что при этом имеется в виду именно прикладная математика.

Все это позволяет обоснованно зафиксировать крайне важный для дальнейшего изложения тезис: методы прикладной математики являются фундаментом дисциплины «Информатика». Последняя, в свою очередь, является прародительницей дисциплин ИС, ИТ и ТП.

Итак, очевидно, что в порожденных диверсификацией условиях неопределенности необходимо использование «обратного» подхода к построению стандартов, когда соответствующие инженерные компоненты вносятся в стандарты, предусматривающие полноценную математическую подготовку. Этот подход, во-первых, идеально обеспечивает фундаментальность образования и гибкость стандартов и, во-вторых,

позволяет решать Задачу путем адаптации существующих, широко распространенных в университетах стандартов.

Несмотря на то что важность математической компетенции выпускника неоднократно отмечена в [1–7], там не сформулирована задача формирования у выпускника именно целостной математической культуры (МК), которая, на наш взгляд, является важнейшим требованием к стандартам ИКТ. Раскроем этот тезис.

А) В [1–7] подчеркивается необходимость формирования у выпускника творческого, абстрактного и даже «в целом математического» [7] мышления. Но ведь именно такое самостоятельное, быстрое, гибкое, строгое аналитическое и логическое мышление и формируется у студента в процессе освоения МК. Как следствие, он способен формулировать и решать самые разнообразные сложные задачи, что также является важнейшим компонентом квалификации специалиста по ИКТ [24, раздел 4.1 и ссылки из него; 3–7].

Б) Основы МК являются универсальным инструментом как текущей профессиональной деятельности, так и самообучения. Владея ими, выпускник будет быстро осваивать любые современные и будущие концепции, методы и технологии ИКТ, что имеет решающее значение в условиях наблюдаемых быстрых изменений ИКТ. Отметим также, что владение МК позволит студенту быстрее осваивать инженерные компоненты ИКТ.

В) Неясен прогноз развития процесса диверсификации в ИКТ (как и прогноз потребности в специалистах по ИКТ с высшим образованием). Одним из весьма вероятных представляется сценарий, когда за счет монополизации и концентрации исследований и разработок произойдут существенные упрощения архитектур и интерфейсов прикладных, прежде всего административных, систем и инфраструктур, а процесс диверсификации сменится на обратный. В этом случае фундаментально подготовленный выпускник, владеющий МК, имеет гораздо больше шансов найти работу в отраслях, отличных от ИКТ.

Г) МК является абсолютно необходимой базой при воспроизводстве кадров высшей квалификации (кандидатов и докторов наук) – преподавателей, исследователей и администраторов, способных к генерации и реализации инновационных идей и необходимых как высшей школе, так и промышленности. Заметим, что схема «бакалавр – магистр» существенно облегчает аспирантам задачу подготовки высококачественной диссертации, обеспечивая увеличение срока работы над ней, при соответствующей организации, с трех до почти пяти лет. Последнее крайне важно для решения все более актуальной задачи восстановления кадрового потенциала высшей школы и науки России.

Далее мы рассмотрим, насколько можно адаптировать к решению Задачи семейство стандартов 01050{0|1} – «Прикладная математика и информатика», 13-летний опыт применения которого имеется в ПетрГУ. Проблемы использования для этой цели других стандартов выходят за рамки данной статьи.

### **Характеристика семейства стандартов**

#### **01050{0|1} – «Прикладная математика и информатика»**

В России существует семейство государственных образовательных стандартов [25–27] «Прикладная математика и информатика» с подготовкой по специальности 010501 (специалисты – 5 лет) и направлениям 010500 (бакалавры – 4 года, магистры – 6 лет). Направление подготовки бакалавров 010500 было утверждено Комитетом по высшей школе в 1992 г., первая, временная, версия стандарта принята в 1993 г., и следующие его версии принципиально от нее не отличаются.

Стандарт имеет следующие основные характеристики. Общий объем работы студента 7 314 часов за 132 учебные недели (8 семестров), или около 73 условных семестровых курса (УСК, примерно соответствует 2 ч. лекций + 2 ч. практических занятий + 2.12 ч. самостоятельной работы в неделю в течение 16.5 недель

одного семестра). Ниже приведено разбиение курсов стандарта по блокам. Первая цифра – процент времени в блоке от общего количества, вторая – количество УСК в блоке:

- гуманитарные и социально-экономические – 24 %, 17;
- естественно-научные – 9 %, 6;
- общие математические – 28 %, 21;
- прикладной математики – 16 %, 12;
- ИКТ общие – 12 %, 9;
- ИКТ по решению факультета / по выбору студента – 11 %, 8.

В курсы блока ИКТ мы включили непосредственно относящиеся к ним курсы из естественно-научного и общепрофессионального циклов. Время, предусмотренное для цикла специальных дисциплин, а также курсов по выбору, региональных и факультативных, мы поделили поровну между блоками прикладной математики и ИКТ.

Таким образом, в рамках стандарта на изучение ИКТ можно выделить около 1 700 часов, эквивалентных 17 УСК, что составляет 23 % времени, которое за счет допустимых в стандарте вариаций может быть увеличено при необходимости до 30 % (2 190 часов, 22 УСК).

В стандарте 010501 подготовки специалистов предусмотрен общий объем работы студента 8 032 часа в девяти семестрах, десятый семестр отводится для практики на предприятиях и разработки дипломной работы. Содержание и структура стандарта практически совпадают со стандартом бакалавра 010500, а объем работы увеличен относительно последнего на 718 часов (примерно 7 УСК). Из них 10 часов приходится на национально-региональный компонент, 20 – на дисциплины по выбору и 688 – на дисциплины специализаций. Это означает, что пятый год обучения позволяет увеличить время на изучение ИКТ до 2 418 часов, эквивалентных 24 УСК (30 % учебного времени).

В стандарте 010500 подготовки магистров предусмотрен общий объем работы магистранта 4 100 часов за 88 учебных недель в течение 2 лет после получения степени бакалавра. Гибкость стандарта обеспечивают двенадцать различных направлений подготовки (проблемных полей), три из которых непосредственно относятся к ИКТ. Это «510209. Математическое и программное обеспечение вычислительных машин», «510210. Программное обеспечение вычислительных сетей» и «510211. Системное программирование». Стандарт имеет следующее разбиение по блокам дисциплин:

- обязательные – 27 % (из них 10 % – региональный компонент);
- специализация – 20 %;
- по выбору – 5 %;
- научно-исследовательская работа – 27 %;
- работа над диссертацией – 21 %.

Таким образом, 73% учебного времени предназначено для совершенствования магистранта по выбранному направлению специализации.

Эти стандарты, являясь, по сути, руководствами по разработке учебных планов, функционально аналогичны руководствам [2–7] АСМ и обладают следующими важными свойствами, позволяющими решать Задачу:

1. В постоянной части предусмотрено формирование целостной МК, которая является фундаментом компетенции в ИКТ.
2. Предусмотрено достаточное количество времени для обязательных курсов по ИКТ.

3. Объем специальных и альтернативных дисциплин полностью обеспечивает факультетам возможность отражать в учебных планах текущие изменения в сфере ИКТ.
4. Адаптация стандарта к решению Задачи может быть реализована не путем апробации и утверждения нового стандарта, а за счет создания новых специализаций, что процедурно существенно проще.
5. Стандарт хорошо зарекомендовал себя в России. В [10, 11] отмечено, что подготовленные на его базе выпускники, работающие в ИКТ, обладают высокими профессиональными качествами.

Рассмотрим вопрос, достаточно ли учебного времени в стандарте бакалавриата 010500 для размещения обязательных тем (Core – Ядро) совокупностей базовых знаний (Body of Knowledge – СБЗ) дисциплин ИН, ИС, ИТ и ТП, определенных в [3–8].

### **Сравнительные временные характеристики Ядер совокупностей базовых знаний**

Нами было выполнено сравнительное исследование учебного времени, выделенного на Ядра СБЗ дисциплин ИС [5], ИТ [6] и ТП [7] относительно Ядра СБЗ ИН [3, 4]. В табл. 1 представлена общая часть Ядер, а в табл. 2 – специальные части Ядер ИТ и ТП, не входящие в Ядро ИН. Специальная часть Ядра ИС описана ниже. В столбцах «темы» табл. 1 названия тем для ИН приведены полностью, а для ИТ и ТП представлены их символичные обозначения из [6] и [7] соответственно. Такие обозначения не приведены в [5], поэтому мы ввели собственные, которые состоят из определенных в [8] номеров разделов второго уровня, буквенных аббревиатур названий разделов и номеров разделов нижних уровней в скобках. Строки табл. 1 содержат темы и соответствующие количества часов, которые, согласно нашей экспертной оценке, являются общими для Ядер.

В столбцах «часы» приведено количество лекционных часов для ИН [3, 4], ИТ [6] и ТП [7]. Для ИС в [5] общий лекционный объем Ядра определен в количестве 10 курсов по 48 часов в семестре, объем лекций по отдельным темам не приводится, он получен нами на основе экспертной оценки близости описаний тем из детализирующего [5] документа [8]. Для всех четырех дисциплин общий объем учебного времени определен в [3–7] как лекционное время, умноженное на четыре. Для приближения временных характеристик ИН к ТП мы добавили в столбце «часы» для ИН 2 часа к теме PL, 1 час к теме SE.1, 1 час к теме SE.4. и 2 часа к теме SE.6., что не влияет на общую картину. Математическая тема DS в руководствах для ИС и ИТ явно не определена и включена в общую часть их Ядер нами, исходя из общих рекомендаций в [5, 6]. Данные в строке «Лекций в Ядре» для ИС, ИТ и ТП взяты из [5–7].

В табл. 2 за названием темы в скобках следует число часов, предусмотренных руководством, в столбце «часы» – количество часов, оставшееся после переноса части часов в общую часть Ядер. Столбцы табл. 2 не связаны между собой.

СБЗ ИС состоит из трех разделов – информационные технологии (ИТ), концепции организации и управления (ОМС) и теория и разработка систем (TDS). Раздел ИТ практически полностью входит в Ядро СБЗ ИН за исключением незначительного количества тем. Раздел ОМС полностью входит в специальную часть СБЗ ИС, за исключением темы «Профессионал и общество». Раздел TDS посвящен проектированию и реализации информационных систем, но содержит примерно 20 % тем по технологии разработки ПО, входящих в Ядро СБЗ ИН.

Ниже для словосочетания «стандарт бакалавриата 010500» мы будем использовать термин «стандарт Б». Объем учебного времени в этом стандарте, отводимый на изучение ИКТ, примем за 1 920 часов и обозначим как Б–ИКТ.

Таблица 1. Общая часть Ядер совокупностей базовых знаний четырех дисциплин сферы ИКТ

| Информатика                           |                                  |      | Информационные системы          |      | Информационные технологии |      | Технология разработки программного обеспечения |      |  |
|---------------------------------------|----------------------------------|------|---------------------------------|------|---------------------------|------|------------------------------------------------|------|--|
| №                                     | темы                             | часы | темы                            | часы | темы                      | часы | темы                                           | часы |  |
| 1                                     | DS. Дискретные структуры         | 43   | включено                        | 43   | включено                  | 43   | FND.mf.1 – 6                                   | 43   |  |
| 2                                     | PF. Основы программирования      | 38   | 1.3 PL (1,3,4), 1.2 ADS (1–3,8) | 38   | PF                        | 38   | CMP.cf.1,3                                     | 38   |  |
| 3                                     | AL. Алгоритмы и теория сложности | 31   | 1.2. ADS (1.2.6–7)              | 31   |                           | 0    | CMP.cf.2                                       | 31   |  |
| 4                                     | AR. Архитектура ЭВМ              | 36   | 1.1 Comp(1,3,4,6)               | 36   | PT2                       | 3    | CMP.cf.5                                       | 26   |  |
| 5                                     | OS. Операционные системы         | 18   | 1.4 OS (1–5)                    | 18   | PT1, PT3, SA1             | 15   | CMP.cf.10                                      | 18   |  |
| 6                                     | NC. Сети ЭВМ                     | 15   | 1.5 Tele (1–5,7,9)              | 15   | IAS5,NET1, WS4            | 16   | CMP.cf.12                                      | 2    |  |
| 7                                     | PL. Языки программирования       | 23   | 1.3 PL (4–7)                    | 23   |                           | 0    | CMP.cf.9, cf.13, ct9                           | 23   |  |
| 8                                     | HC. Интерфейс человек ЭВМ        | 8    | 1.4 OS (10), 3.9 ISD (6)        | 8    | HCI1, HCI4                | 9    | DES.hci.1 – 10                                 | 8    |  |
| 9                                     | GV. Графика и визуализация       | 3    | 3.9 ISD(6)                      | 3    |                           | 0    | в спец. областях                               | 0    |  |
| 10                                    | IS. Интеллектуальные системы     | 10   | 1.7 AI                          | 10   |                           | 0    | в спец. областях                               | 0    |  |
| 11                                    | IM. Управление информацией       | 10   | 1.6 DB (1-3), 3.3 SDCM (2)      | 10   | IM1                       | 8    | CMP.cf.11                                      | 10   |  |
| 12                                    | SP. Профессионал и общество      | 16   | 2.9 Pro (7), 2.8 LEA (1–7)      | 16   | SP2 ,3, 5, 8,9            | 11   | PRF.pr.1-4, 6                                  | 16   |  |
| 13                                    | SE. Технология разработки ПО     | 35   |                                 | 27   |                           | 0    |                                                | 35   |  |
|                                       | SE1. Проектирование              | 9    | 3.4 SDTT                        | 9    |                           |      | DES.con, ES.str                                | 9    |  |
|                                       | SE2. Использование API           | 5    |                                 | 0    |                           |      | CMP.ct.1                                       | 5    |  |
|                                       | SE3. Инструменты и среды         | 3    | 3.2 ASD (1)                     | 3    |                           |      | CMP.tl.1-5                                     | 3    |  |
|                                       | SE4. Процесс разработки          | 3    | 3.3 SDCM (1)                    | 2    |                           |      | PRO.con                                        | 3    |  |
|                                       | SE5. Спецификации и требования   | 4    | 3.9 ISD (7)                     | 4    |                           |      | MAA.rfd.1-10                                   | 4    |  |
|                                       | SE6. Проверка соответствия       | 5    | 3.9 ISD (7)                     | 3    |                           |      | VAV.fnd                                        | 5    |  |
|                                       | SE7. Эволюция                    | 3    | 2.2 ISM (15)                    | 3    |                           |      | EVO.pro.1 – 3                                  | 3    |  |
|                                       | SE8. Управление проектами        | 3    | 3.10 SITS (7)                   | 3    |                           |      | MGT.con.1-5.                                   | 3    |  |
| Итого лекций в общей части Ядер       |                                  | 286  |                                 | 278  |                           | 143  |                                                | 250  |  |
| Без математической темы DS            |                                  |      |                                 |      |                           |      |                                                |      |  |
| Лекций в общей части Ядер             |                                  | 243  |                                 | 235  |                           | 100  |                                                | 207  |  |
| Лекций в специальных частях           |                                  | нет  |                                 | 245  |                           | 181  |                                                | 244  |  |
| Лекций в Ядре                         |                                  | 243  |                                 | 480  |                           | 281  |                                                | 451  |  |
| Учебного времени в специальных частях |                                  | нет  |                                 | 980  |                           | 724  |                                                | 976  |  |
| Учебного времени в Ядре               |                                  | 972  |                                 | 1920 |                           | 1124 |                                                | 1804 |  |

Таблица 2. Специальные части Ядер совокупностей базовых знаний

| Информационные технологии                           |      | Технология разработки программного обеспечения    |      |
|-----------------------------------------------------|------|---------------------------------------------------|------|
| темы                                                | часы | темы                                              | часы |
| ITF. Основы ИТ (33)                                 | 33   | CMP. Основы ИКТ (172)                             | 17   |
| HCI. Интерфейс человек ЭВМ (20)                     | 11   | FND. Математические и технологические основы (89) | 46   |
| IM. Управление информацией (34)                     | 26   | PRF. Практика в профессии (35)                    | 19   |
| IAS. Достоверность и безопасность данных (23)       | 21   | MAA. Моделирование и анализ ПО (53)               | 49   |
| IPT. Интегрирующие технологии программирования (23) | 23   | DES. Проектирование ПО (45)                       | 28   |
| NET. Сети ЭВМ (20)                                  | 9    | VAV. Верификация и проверка соответствия ПО (42)  | 37   |
| PT. Технологии платформ (14)                        | 0    | EVO. Эволюция ПО (10)                             | 7    |
| SA. Администрирование и эксплуатация систем (11)    | 7    | PRO. Процесс разработки ПО (13)                   | 10   |
| SIA. Системная интеграция и архитектура (21)        | 21   | QUA. Качество ПО (16)                             | 16   |
| SP. Профессионал и общество (23)                    | 12   | MGT. Управление ПО (19)                           | 15   |
| WS. Web системы и технологии (21)                   | 18   |                                                   |      |
| Итого                                               | 181  | Итого                                             | 244  |

Анализ полученных временных характеристик показывает, что знания Ядра ИН почти полностью включены в Ядра ИС, ТП и ИТ. При этом объем общих тем Ядра ИТ существенно меньше за счет отсутствия тем AL, PL, GV, IS, SE и существенно меньшего объема темы AR. Это означает, что стандарт Б, с учетом естественного включения темы DS в блок общих математических курсов, позволяет свободно разместить в Б–ИКТ общий объем учебного времени Ядер СБЗ каждой из четырех дисциплин.

Таким образом, мы приходим к основополагающему заключению: знания Ядра ИН являются базовыми знаниями сферы ИКТ в целом, т. е. являются фундаментальными для дисциплин ИС, ИТ и ТП, что позволяет считать последние прикладными дисциплинами по отношению к ИН. Это заключение хорошо согласуется с разделами [4–7], посвященными связям Ядер ИС, ИТ и ТП с Ядром ИН.

Анализ табл. 1 и 2 позволяет сделать следующие выводы о соотношениях общего учебного времени, предусмотренного в стандарте Б и в руководствах [3–7], для дисциплин ИН, ИС, ИТ и ТП.

А) В [3, 4] отмечено, что общий объем учебного времени по ИН в университетах США и Канады сильно варьируется и конкретный объем его не приводится. Если вычислить его, полагая, что всего выпускник должен, кроме курсов Ядра, изучить 12 курсов ИН по 160 часов (без учета темы DS), то это дает 237 часов лекционного времени в специальной части дополнительно к 243 часам Ядра, т. е. всего 480 часов лекций или 1 920 часов учебного времени. Это означает, что весь объем учебного времени ИН (не только СБЗ) полностью размещается в Б–ИКТ, особенно если учесть, что в его математических блоках предусмотрены курсы, которые в [3, 4, раздел 9.3] (см. также раздел 1 данной статьи) рекомендуется включать в учебные планы как углубленные. Таким образом, стандарт Б можно считать эквивалентным руководству [3, 4] как по содержанию, так и по объему знаний по ИН.

Б) Весь объем учебного времени ИТ (1 800 часов лекций) не только полностью размещается в Б–ИКТ, но и позволяет (при использовании стандарта Б для подготовки по ИТ) дать выпускникам рекомендуемую математическую подготовку [6, раздел 8.1.1] и продвинутые курсы [6, раздел 8.2].

В) Объем Ядра ИС также полностью размещается в Б–ИКТ. В [5] отмечено, что это Ядро рассчитано на изучение в течение двух лет, а рекомендации по планированию полного объема учебного времени приво-

дятся в общей форме: «Темы, необходимые для учебного плана ИС, включают... дискретную математику, введения в математический анализ и статистику... принципы экономики, бухгалтерское и финансовое дело, человеческие ресурсы, маркетинг и логистику». При использовании стандарта Б для подготовки специалистов по ИС изучение математических курсов происходит естественным образом, а экономические и организационные курсы, не входящие в Ядро ИС, можно давать за счет времени гуманитарного блока и частично блока прикладной математики.

Г) Объем Ядра ТП полностью размещается в Б–ИКТ. При этом дополнительные курсы, которые рекомендованы в [7] для включения в полный учебный план («Математический анализ», «Физика», гуманитарные и социальные курсы), входят также и в стандарт Б. Таким образом, значительная часть полного объема учебного времени по ТП также может быть размещена в стандарте Б.

Вопрос возможности размещения полного объема учебного времени дисциплин ИС и ТП в стандарте Б выходит за рамки данной статьи и требует дополнительного исследования. В то же время наш анализ однозначно показывает, что стандарт Б можно использовать для обучения как знаниям Ядер СБЗ, так и значительной части знаний полных учебных планов всех четырех дисциплин. При этом можно с уверенностью сказать, что стандарт специальности 010501 (5 лет обучения) позволяет готовить выпускников по всем 4 дисциплинам в полном соответствии с рекомендациями [3–8], а стандарт магистратуры 010500 обеспечивает углубленное изучение дисциплин и подготовку к обучению в аспирантуре.

Адаптацию стандарта Б к подготовке выпускников по четырем дисциплинам ИКТ следует проводить путем фиксации в нем трех профессиональных блоков: Математического Ядра, Ядра ИКТ и специальных блоков, соответствующих каждой из дисциплин. Содержание блоков можно считать сформированным, и задача состоит в разработке разумных связей между ними. С процедурной точки зрения в стандарт Б достаточно ввести четыре специализации ИН, ИС, ИТ и ТП.

### **Опыт использования стандарта подготовки бакалавров 010500 на математическом факультете Петрозаводского государственного университета**

В ПетрГУ бакалавратура по направлению 010500 (510200) была открыта в 1993 г. [28–31], магистратура – в 1997 г. Учебный план был разработан с учетом рекомендаций Computing Curricula 1991. За тринадцать лет факультет выпустил 269 бакалавров, 191 специалиста и 73 магистра. Характерно, что только единичные студенты покидали университет, получив диплом бакалавра, как правило, для продолжения образования в других университетах, в том числе за границей. Большинство студентов продолжали обучение в течение одного или двух лет, получая диплом специалиста или магистра. Часто магистранты продолжали образование в аспирантуре.

Наш опыт подтверждает, что стандарт Б обеспечивает исключительную гибкость, позволяющую отражать в учебных планах текущие изменения в сфере ИКТ. За прошедшие 13 лет мы последовательно ввели в учебный план курсы «Введение в процессоры», «Компьютерные сети», «Операционные системы», «Технология разработки программного обеспечения», «Язык процессов shell», «Объектно-ориентированное программирование в средах Java и .NET», «Архитектура ЭВМ», «Технологии Web», «Групповой проект по ТП».

За счет курсов по выбору и факультативных в 2001 г. было открыто специализированное направление подготовки «Системные сетевые технологии», на котором читаются курсы «Взаимодействующие системы (Concurrent Systems)», «Программирование в ОС Unix», «Сетевое программирование», «Распределенные системы».

Учебные планы на основе стандарта Б легко адаптируются к международным требованиям, что подтверждается нашим опытом по разработке общего ядра учебных планов с отделением информатики (Computer Science) Хельсинкского университета (Финляндия) [32].

В настоящее время факультет использует следующую последовательность вводных курсов ИКТ (математические курсы не представлены):

| 1-й семестр                                      | УСК | 2-й семестр                                             | УСК  |
|--------------------------------------------------|-----|---------------------------------------------------------|------|
| Введение в ИКТ                                   | 0.5 | Язык процессов shell                                    | 0.5  |
| Введение в процессоры (язык ASM)                 | 0.5 | Введение в процессоры (язык ASM)                        | 0.5  |
| Введение в программирование и алгоритмы (язык C) | 1   | Алгоритмы, структуры данных и введение в ООП (язык C++) | 1    |
| Дискретная математика                            | 1   | Математическая логика                                   | 0.75 |

| 3-й семестр             |      | 4-й семестр             |      |
|-------------------------|------|-------------------------|------|
| Комбинаторные алгоритмы | 1    | Комбинаторные алгоритмы | 1    |
| Базы данных             | 1    | Операционные системы    | 0.75 |
| Компьютерные сети       | 0.75 | ООП в среде .NET        | 0.75 |
| ООП в среде Java        | 0.75 |                         |      |

Идея этой последовательности – параллельное ознакомление с архитектурой процессора и основными стандартными приемами процедурного программирования на языке C. При этом используется разработанная нами методика преподавания курса «Введение в процессоры» для первокурсников с разработкой простых программ на языке ассемблера для реальных процессоров [33, 34]. Такой подход, в частности, позволяет избежать серьезных трудностей, возникающих при использовании в первом курсе по программированию объектных языков (например, Java), которые детально анализируются в [35].

Мы опираемся на языки C и C++, потому что они являются прародителями большинства современных языков, активно используются в индустрии ПО и, по сути, являются основой современной культуры программирования и средством общения в профессиональных сообществах. На базе культуры C легко строить курсы по методам ООП, операционным системам, сетевым технологиям, компиляторам компиляторов и другим базовым курсам ИКТ.

## Заключение

Мы рассмотрели проблему разработки и внедрения образовательных стандартов для решения задачи подготовки специалистов, способных к долговременной эффективной профессиональной работе в сфере информационных и коммуникационных технологий в условиях ее диверсификации и быстрых значительных изменений. В статье предлагается удовлетворять растущую потребность в таких специалистах, в том числе и за счет адаптации существующих, широко распространенных стандартов, что существенно сокращает время реакции высшей школы на изменения содержания и потребностей сферы ИКТ.

Исходя из тезиса, что методы прикладной математики являются фундаментом дисциплины «Информатика», мы подчеркиваем важность математической культуры для всех дисциплин ИКТ и предлагаем «обратный» подход к построению стандартов, когда соответствующие инженерные компоненты вносятся в стандарты, предусматривающие усиленную математическую подготовку. При этом в качестве базовых стандартов подготовки естественно использовать семейство стандартов 01050{0|1} – «Прикладная математика и информатика», которое широко распространено в университетах России, давно и успешно апробировано, а его выпускники имеют хорошую профессиональную репутацию.

Выполненный нами сравнительный анализ временных характеристик Ядер совокупностей базовых знаний дисциплин «Информатика», «Информационные системы», «Информационные технологии» и «Технология разработки программного обеспечения» показывает, что эти Ядра полностью размещаются в объеме учебного времени стандарта бакалавриата 010500, выделенного на изучение ИКТ. При этом адаптация

стандарта к современным требованиям ИКТ процедурно может быть выполнена за счет введения четырех специализаций, соответствующих дисциплинам ИН, ИС, ИТ и ТП.

Подход, основанный на адаптации семейства 01050{0|1} к современным потребностям и состоянию сферы ИКТ, имеет следующие преимущества:

- направленное формирование целостной математической культуры;
- апробированная двухступенчатая схема «бакалавр – магистр»;
- гибкая структура, обеспечивающая возможность реакции на изменения;
- легкая адаптация к международным требованиям;
- процедурная простота процесса адаптации по сравнению с процессом разработки, апробации и введения новых стандартов.

Опыт использования семейства 01050{0|1} на математическом факультете ПетрГУ в течение 13 лет подтверждает представленные в статье тезисы.

### Благодарности

Автор признателен администрации ПетрГУ в лице долгое время возглавлявшего ПетрГУ ректора В.Н. Васильева и действующего ректора А.В. Воронина, проректора Н.С. Рузановой и бывшего первого проректора А.А. Печникова, декана В.А. Шестакова за постоянное внимание к проблемам, рассмотренным в статье и неизменную поддержку. Написание этой статьи было бы невозможно без крайне полезных и заинтересованных обсуждений, в которых участвовали мои коллеги Т. Аланко, О.Ю. Богоявленская, А.Г. Варфоломеев, В.Т. Вдовицын, А.В. Воронин, А.В. Иванов, С.Т. Коржов, Д.Ж. Корзун, В.А. Кузнецов, А.А. Печников, В.В. Поляков, В.А. Пономарев, А.А. Рогов, Г.С. Сиговцев, А.В. Соколов, В.В. Старков, Л.В. Щеголева. Всем им автор выражает глубокую благодарность.

### Библиографический список

1. IEEE/AIS/ACM Joint Task Force on Computing Curricula. Computing Curricula 2005. The Overview Report covering undergraduate degree programs in Computer Engineering, Computer Science, Information Systems, Information Technology, Software Engineering. 2005  
(<http://www.computer.org/curriculum> или <http://www.acm.org/education/curricula.html>).
2. IEEE/ACM Joint Task Force on Computing Curricula. Computer Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering. IEEE Computer Society Press and ACM Press, 2004  
(<http://www.computer.org/curriculum> или <http://www.acm.org/education/curricula.html>).
3. ACM/IEEE–Curriculum 2001 Task Force. Computing Curricula 2001, Computer Science. IEEE Computer Society Press and ACM Press, 2001 (<http://www.computer.org/curriculum> или <http://www.acm.org/education/curricula.html>).
4. Рекомендации по преподаванию информатики в университетах / Ред. В.Л. Павлов, А.А. Терехов. СПб.: Изд-во СПбГУ, 2002. С. 367 (русский перевод [3]).
5. ACM/AIS/AITP Joint Task Force on Information Systems Curricula. IS2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, Association for Computing Machinery, Association for Information Systems, and Association for Information Technology Professionals. 2002  
(<http://www.acm.org/education/curricula.html> или <http://www.computer.org/curriculum>).
6. The ACM SIGITE Task Force on IT Curriculum. Information Technology, Computing Curricula Information Technology Volume. Curriculum Guidelines for Undergraduate Degree Programs in Information Technology  
(<http://www.acm.org/education/curricula.html>).

7. IEEE/ACM Joint Task Force on Computing Curricula. Software Engineering 2004, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. IEEE Computer Society Press and ACM Press, 2004 (<http://www.computer.org/curriculum> или <http://www.acm.org/education/curricula.html>).
8. Detailed Body of Information Systems Knowledge (<http://192.245.222.212:8009/IS2002reportsPDF/rptBodyOfKnowledge.pdf>).
9. Терехов А.Н. Как готовить системных программистов / А.Н. Терехов // Компьютерные инструменты в образовании. 2001. № 3–4. С. 2–80.
10. Сухомлин В.А. «Информационные технологии» – актуальное образовательное направление / В.А. Сухомлин // Информационные технологии. 2002. № 8. С. 9–17.
11. Ивановский С.А. Профессиональная подготовка программистов в рамках специальностей и направлений государственных образовательных стандартов: Доклад на II конференции «Преподавание информационных технологий в Российской Федерации» / С.А. Ивановский, А.Р. Лисс, В.В. Романцев, А.В. Экало. 2004 (<http://www.it-education.ru/2004/reports/romantsev.htm>).
12. Материалы конференций «Преподавание информационных технологий в Российской Федерации», 2003–2006 (<http://www.it-education.ru/2006>).
13. Сухомлин В.А. ИТ-образование: концепция, образовательные стандарты, процесс стандартизации / В.А. Сухомлин. М.: Горячая линия – Телеком, 2005.
14. Никитин В.В. Совершенствование структуры и стандартов системы профессионального ИКТ-образования: Доклад на III конференции «Преподавание информационных технологий в Российской Федерации». 2005 ([http://www.it-education.ru/2005/reports/1\\_Nikitin.htm](http://www.it-education.ru/2005/reports/1_Nikitin.htm)).
15. Труды первой международной научно-практической конференции «Современные информационные технологии и ИТ-образование». М.: Изд-во МГУ, 2005.
16. Государственный образовательный стандарт высшего профессионального образования. Специальность 010503 «Математическое обеспечение и администрирование информационных систем», Квалификация математик-программист / Министерство образования РФ. М., 2000.
17. Государственный образовательный стандарт высшего профессионального образования. Направление 010300 «Математика, компьютерные науки». Степень – бакалавр математики / Министерство образования РФ. М., 2000.
18. Государственный образовательный стандарт высшего профессионального образования. Направление 010400 «Информационные технологии». Степень – бакалавр информационных технологий / Министерство образования РФ. М., 2002.
19. Государственный образовательный стандарт высшего профессионального образования. Направление 010400 «Информационные технологии». Степень – магистр информационных технологий (магистр науки) / Министерство образования РФ. М., 2002.
20. Государственный образовательный стандарт высшего профессионального образования. Направление 080700 «Бизнес-информатика», Степень (квалификация) – бакалавр бизнес-информатики / Министерство образования РФ. М., 2005.
21. Ассоциация предприятий компьютерных и информационных технологий (<http://www.apkit.ru>).
22. Андрианов С.Н. О ходе подготовки бакалавров по направлению «Информационные технологии» на факультете ПМ–ПУ СПбГУ / С.Н. Андрианов, Е.И. Веремей // Труды первой международной научно-практической конференции «Современные информационные технологии и ИТ-образование».
23. Решение конференции «Преподавание информационных технологий в Российской Федерации». 2005 (<http://www.it-education.ru/2005>).

24. Goldweber M. Historical perspectives on the computing curriculum (Report of WG № 7), New-York, USA, in Working Group Reports and Supplemental Proceedings of ITiCSE'97 / M. Goldweber, J. Impagliazzo, A.G. Clear, G. Davies, I.A. Bogoiavlenski, H. Flack, J.P. Mayers, R. Rasala. Uppsala (Sweden): ACM Press, 1997. P. 94–111.
25. Государственный образовательный стандарт высшего профессионального образования. Направление 010500 «Прикладная математика и информатика». Степень – бакалавр прикладной математики и информатики / Министерство образования РФ. М., 2000.
26. Государственный образовательный стандарт высшего профессионального образования. Специальность 010501 – Прикладная математика и информатика. Квалификация – математик, системный программист / Министерство образования РФ. М., 2000.
27. Государственный образовательный стандарт высшего профессионального образования. Направление 010500 «Прикладная математика и информатика». Степень – магистр прикладной математики и информатики / Министерство образования РФ. М., 2000.
28. Сиговцев Г.С. О проекте профессионально-образовательной программы для бакалавров по специальности «Информатика» (Computer Science) / Г.С. Сиговцев, А.А. Печников., Ю.А. Богоявленский // Тезисы докладов международного семинара «Двухступенчатое образование и самофинансирование вузов». Юрмала: Балтик Легис Интернешнл, 1992.
29. Воронин А.В. Профессионально-образовательная программа для подготовки бакалавров по направлению E02 «Прикладная математика и информатика» / А.В. Воронин, Ю.А. Богоявленский, В.А. Кузнецов, В.В. Поляков, Г.С. Сиговцев // Тезисы докладов Всероссийской научно-методической конференции «Методические основы функционирования и развития системы многоуровневого образования». Саратов: Изд-во Саратовского гос. тех. ун-та, 1993.
30. Печников А.А. Программа подготовки бакалавров в ПетрГУ по направлению E02 «Прикладная математика и информатика» / А.А. Печников, Ю.А. Богоявленский, А.В. Воронин, В.А. Кузнецов, В.В. Поляков, Г.С. Сиговцев. Петрозаводск: Изд-во ПетрГУ, 1994. С. 75–80 (Сер. «Прикладная математика и информатика». Т. 3).
31. Bogoiavlenski I. Using of Computing Curricula 1991 for Transition from «Mathematics» to «Applied Mathematics and Computer Science» Baccalaureate Program / I. Bogoiavlenski, A. Pechnikov, G. Sigovtsev, A. Voronin // Abstracts of Conference ITiCSE'97. Uppsala: University of Uppsala, 1997. P. 8.
32. Bogoyavlenskiy Y., Alanko T. Common Core of Working Study Program in Computer Science ([http://www.cs.karelia.ru/fdpw/2003/YBGV/IOuri\\_Common-Core.pdf](http://www.cs.karelia.ru/fdpw/2003/YBGV/IOuri_Common-Core.pdf)).
33. Bogoiavlenski I. Five Years Experience of Architecture and Assembly Language Introduction Course for First Year Students, Proceedings of the Interdisciplinary Workshop on Complex Learning in Computer Environment (CLCE'94) / I. Bogoiavlenski, A. Pechnikov. University of Joensuu (Finland). Report P8-1994-04, 1994. P. 124–128.
34. Богоявленский Ю.А. Центральные процессоры Intel 8086/8088. Архитектура и система команд / Ю.А. Богоявленский, А.А. Печников. Петрозаводск: Изд-во ПетрГУ, 1992, 1997, 2000.
35. Bruce Kim B. Controversy on How to Teach CS 1: A Discussion on the SIGCSE-members Mailing List // Inroads – SIGCSE Bulletin. 2004. Vol. 36. № 4. December. P. 29–34.

## EDUCATION OF INFORMATION-COMMUNICATION TECHNOLOGY ON THE BASE OF RUSSIAN STATE EDUCATIONAL STANDARD «APPLIED MATHEMATICS AND INFORMATICS»

**Y.A. Bogoyavlenskiy**

Petrozavodsk State University

Petrozavodsk, Russia

ybgv@cs.karelia.ru

The area of Information-Communication Technology (ICT) is expanding very fast, enormously huge funds are being invested in its development. New paradigms, concepts, standards, tools, applications are rapidly appearing and introducing into industry and society. The ICT development drastically increases both importance and complexity of the problem of creating adequate ICT Curricula educational standards (models) to teach professionals in such a way, that they should be capable of long duration and effective activity in the ICT area.

The problem is in the focus of interest of teachers and administrators community which is illustrated by recent intensive activity of task forces of ACM, IEEE-CS, AIS, SIGITE (SITE), BCS, IFIP, ABET/CAC/ CSAB aimed of develop adequate Curricula models for ICT specialties family.

The problem in Russia may be solved at the account of well-established tradition of providing education fundamentality. The essence of the tradition is, that a student learns not set of means for solving concrete problems, but theories, methods and approaches underlying of the development of the means. Fundamentality of the qualification firstly allows organizing manufacturing a product of good quality with high labour productivity, and, secondly, provides for a graduate self-teaching ability and his/her fast adaptation to any innovations in the area.

In Russia exists a family of state educational standards «Applied Mathematics and Informatics» for education by specialty 010200 (5 years study) and by directions 510200 (bachelor, 4 year, and master of science, 6 year). The standards, per se, are curricula models, as is, for example, «Computing Curricula 2001», and they have three important features, allowing to organize on their base education of modern ICT professionals.

Firstly, serious mathematical training forms way of student's thinking, which is fast, flexible and strictly analytical. Secondly, there enough place in the standard, to give students basics of Computing. Thirdly, predefined in the standard volume of time for special and elective disciplines fully provides planning officers ability to reflect in a curriculum current changes in the ICT area.

Remark, that existing big positive experience of using the graduates, educated by the standards.

Using 13 years experience of Mathematical Faculty of the Petrozavodsk State University in education of bachelors of 510200 direction we will present the paper covering the following topics: expanded motivation of using {015}10200 family of standards for education high qualified ICT professionals; propositions on content of main knowledge areas; bachelor degree Curriculum model; main direction of specializations.

## ТРОИЧНЫЕ ЭВМ «СЕТУНЬ» И «СЕТУНЬ 70»

**Н.П. Брусенцов, Рамиль Альварес Хосе**

Московский государственный университет

Москва, РФ

ramil@cs.msu.ru

В начале 1956 г. по инициативе академика С.Л. Соболева, заведующего кафедрой вычислительной математики на механико-математическом факультете Московского университета, в вычислительном центре

МГУ был учрежден отдел электроники и стал работать семинар с целью создать практичный образец цифровой вычислительной машины, предназначенной для использования в вузах, а также в лабораториях и конструкторских бюро промышленных предприятий. Требовалось разработать малую ЭВМ, простую в освоении и применении, надежную, недорогую и вместе с тем эффективную в широком спектре задач.

Обстоятельное изучение в течение года имевшихся в то время вычислительных машин и технических возможностей их реализации привело к нестандартному решению употребить в создаваемой машине не двоичный, а троичный симметричный код, реализовав ту самую «уравновешенную» систему счисления, которую Д. Кнут двадцать лет спустя назовет «быть может, самой изящной» [1] и как затем стало известно, достоинства которой были выявлены К. Шенноном в 1950 г. [2].

В отличие от общепринятого в современных компьютерах двоичного кода с цифрами 0, 1, арифметически неполноценного вследствие невозможности непосредственного представления в нем отрицательных чисел, троичный код с цифрами  $-1$ , 0, 1 обеспечивает оптимальное построение арифметики чисел со знаком. При этом не только нет нужды в искусственных и несовершенных дополнительном, прямом либо обратном кодах чисел, но арифметика обретает ряд значительных преимуществ: единообразие кода чисел, варьируемая длина операндов, единственность операции сдвига, трехзначность функции «знак числа», оптимальное округление чисел простым отсечением младших разрядов, взаимокompенсированность погрешностей округления в процессе вычисления [3].

Троичная ЭВМ «Сетунь» [4, 5], опытный образец которой разработали, смонтировали и к концу 1958 г. ввели в эксплуатацию сотрудники отдела электроники, как показал опыт ее освоения, программного оснащения и многообразных практических применений, с исчерпывающей полнотой удовлетворяла всем предусмотренным заданием на ее разработку требованиям. Этот успех, с учетом того, что разработка троичной ЭВМ предпринималась впервые, проводилась немногочисленным коллективом начинающих сотрудников (8 выпускников МЭИ и МГУ, 12 техников и лаборантов) и была выполнена в короткий срок, явно свидетельствует о благодатности троичной цифровой техники. Ценой усложнения по сравнению с двоичными элементов памяти и элементарных операций достигается существенное упрощение и, главное, естественность архитектуры троичных устройств.

При минимальном наборе команд (всего 24 одноадресные команды) «Сетунь» обеспечивала возможность вычислений с фиксированной и с плавающей запятой, обладала индекс-регистром, значение которого можно как прибавлять, так и вычитать при модификации адреса, предоставляла операцию сложения с произведением, оптимизирующую вычисление полиномов, операцию поразрядного умножения и три команды условного перехода по знаку результата. Простая и эффективная архитектура позволила усилиями небольшой группы программистов уже к концу 1959 г. оснастить машину системой программирования и набором прикладных программ [6], достаточными для проведения в апреле 1960 г. междуведомственных испытаний опытного образца.

По результатам этих испытаний «Сетунь» была признана «первым действующим образцом универсальной вычислительной машины на безламповых элементах», которому свойственны «высокая производительность, достаточная надежность, малые габариты и простота технического обслуживания». По рекомендации Междуведомственной комиссии Совет Министров СССР принял постановление о серийном производстве «Сетуни» на Казанском заводе математических машин. Но почему-то троичный компьютер пришелся не по нраву чиновникам радиоэлектронного ведомства: они не обеспечили разработку серийного образца машины, а после того как он все-таки был осуществлен с использованием конструктивов выпускавшейся заводом машины М-20, не содействовали наращиванию выпуска в соответствии с растущим числом заказов,

в частности из-за рубежа, а наоборот, жестко ограничивали выпуск, отклоняя заказы, и в 1965 г. полностью прекратили, причем воспрепятствовали освоению машины в ЧССР, планировавшей ее крупносерийное производство. Поводом для этой странной «политики» могла быть рекордно низкая цена «Сетуни» – 27,5 тыс. рублей, обусловленная бездефектным производством ее магнитных цифровых элементов на Астраханском заводе ЭА и ЭП, по 3 руб. 50 коп. за элемент (в машине было около 2 тыс. элементов). Существенно то, что электромагнитные элементы «Сетуни» позволили осуществить пороговую реализацию трехзначной логики на редкость экономно, естественно и надежно. Опытный образец машины за 17 лет эксплуатации в ВЦ МГУ, после замены на первом году трех элементов с дефектными деталями, не потребовал никакого ремонта внутренних устройств и был уничтожен в состоянии полной работоспособности. Серийные машины устойчиво функционировали в различных климатических зонах от Одессы и Ашхабада до Якутска и Красноярска при отсутствии какого-либо сервиса и запчастей.

Благодаря простоте и естественности архитектуры, а также рационально построенной системе программирования, включающей интерпретирующие системы: ИП-2 (плавающая запятая, 8 десятичных знаков), ИП-3 (плавающая запятая, 6 десятичных знаков), ИП-4 (комплексные числа, 8 десятичных знаков), ИП-5 (плавающая запятая, 12 десятичных знаков), автокод ПОЛИЗ с операционной системой и библиотекой стандартных подпрограмм (плавающая запятая, 6 десятичных знаков), машины «Сетунь» успешно осваивались пользователями в вузах, на промышленных предприятиях и в НИИ, оказываясь эффективным средством решения практически значимых задач в самых различных областях от научно-исследовательского моделирования и конструкторских расчетов до прогноза погоды и оптимизации управления предприятием [7]. На семинарах пользователей вычислительных машин «Сетунь», проведенных в МГУ (1965), на Людиновском тепловозостроительном заводе (1968), в Иркутском политехническом институте (1969) были представлены десятки сообщений о результативных народнохозяйственных применениях этих машин. «Сетунь», благодаря естественности троичного симметричного кода, оказалась поистине универсальным, несложно программируемым и весьма эффективным вычислительным инструментом, положительно зарекомендовавшим себя, в частности, как техническое средство обучения вычислительной математике более чем в тридцати вузах. А в Военно-воздушной инженерной академии им. Жуковского именно на «Сетуни» была впервые реализована автоматизированная система компьютерного обучения [8].

Троичная система счисления основана на том же позиционном принципе кодирования чисел, что и принятая в современных компьютерах двоичная система, однако вес  $i$ -й позиции (разряда) в ней равен не  $2^i$ , а  $3^i$ . При этом сами разряды не двухзначны (не биты), а трехзначны (триты) – помимо 0 и 1 допускают третье значение, которым в симметричной системе служит  $-1$ , благодаря чему единообразно представимы как положительные, так и отрицательные числа. Значение  $n$ -тритного целого числа  $N$  определяется аналогично значению  $n$ -битного:

$$N = \sum_{i=0}^{n-1} \alpha_i 3^i = \alpha_{n-1} 3^{n-1} + \alpha_{n-2} 3^{n-2} + \alpha_{n-3} 3^{n-3} + \dots + \alpha_1 3 + \alpha_0,$$

где  $\alpha_i \in \{1, 0, -1\}$  – значение цифры  $i$ -го разряда.

Цифры в троичной симметричной системе целесообразно обозначать их знаками, т.е. вместо 1, 0, -1 писать +, 0, -. Например, десятичные числа 13, 7, 6, -6 в такой троичной записи будут: 13 = +++, 7 = ++-, 6 = +-0, -6 = --0. Изменение знака числа в симметричном коде равносильно потритной инверсии, т.е. взаимоза-

мене всех «+» на «-» и всех «-» на «+». Операции сложения и умножения в троичном симметричном коде определены таблицами:

|   | +   | 0 | -   |
|---|-----|---|-----|
| + | + - | + | 0   |
| 0 | +   | 0 | -   |
| - | 0   | - | - + |

|   | + | 0 | - |
|---|---|---|---|
| + | + | 0 | - |
| 0 | 0 | 0 | 0 |
| - | - | 0 | + |

В отличие от двоичной, это арифметика чисел со знаком, причем знаком числа оказывается цифра старшего из его значащих (ненулевых) разрядов. Проблемы чисел со знаком, не имеющей в двоичном коде совершенного решения, в троичном симметричном коде просто нет, чем и обусловлены его принципиальные преимущества.

Машина «Сетунь» может быть охарактеризована как одноадресная, последовательного действия, с 9-тритным кодом команды, 18-тритными регистрами сумматора  $S$  и множителя  $R$ , 5-тритными индекс-регистром модификации адреса  $F$  и счетчиком-указателем выполняемых команд  $C$ , а также однобитным указателем знака результата  $\omega$ , управляющим условными переходами.

Оперативная память – 162 9-тритных ячейки – разделена на 3 страницы по 54 ячейки для постраничного обмена с основной памятью – магнитным барабаном емкостью 36 либо 72 страницы. Считывание и запись в оперативную память возможны 18-тритными и 9-тритными словами, причем 9-тритное слово соответствует старшей половине 18-тритного в регистрах  $S$  и  $R$ . Содержимое этих регистров интерпретируется как число с фиксированной после второго из старших разрядов запятой, т.е. по модулю оно меньше 4,5. При вычислениях с плавающей запятой мантисса  $M$  нормализованного числа удовлетворяет условию  $0,5 < |M| < 1,5$ , а порядок представлен отдельным 5-тритным словом, интерпретируемым как целое со знаком.

Страничная двухступенная структура памяти с пословной адресацией в пределах трех страниц ОЗУ, обходящейся 5-тритными адресами и соответственно 9-тритными командами, обусловила необыкновенную компактность программ и вместе с тем высокое быстродействие машины, несмотря на то, что в интерпретирующих системах магнитный барабан функционирует как оперативная память.

В 1967–1969 гг. на основе опыта создания и практических применений машины «Сетунь» разработана усовершенствованная троичная цифровая машина «Сетунь 70», опытный образец которой вступил в строй в апреле 1970 г. Это была машина нетрадиционной двухстековой архитектуры, ориентированной на обеспечение благоприятных условий дальнейшего развития ее возможностей методом интерпретирующих систем [9].

Принятие арифметического стека (стека 18-тритных операндов) обусловлено использованием в качестве машинного языка так называемой польской инверсной записи программ (ПОЛИЗ), положительно зарекомендовавшей себя в одноименном интерпретаторе на «Сетуни». ПОЛИЗ-программа состоит не из команд той или иной адресности, а является последовательностью коротких слов – 6-тритных трайтов (троичных байтов). Как элемент программы трайт может быть либо адресным, либо операционным. Адресный трайт либо используется в качестве операнда предшествующим операционным, либо воспринимается как предписание заслат в стек операндов из оперативной памяти адресуемое слово от одного до трех трайтов. В оперативной памяти всего 9 страниц по 81 трайту, причем открыты для доступа в данный момент три страницы, номера которых указаны в так называемых «регистрах приписки».

Операционный трайт указывает операции, а вернее процедуры, выполняемые над стеком операндов, а также над регистрами процессора. Всего предусмотрена 81 операция – 27 основных, 27 служебных и 27 программируемых пользователем.

Второй (системный) стек, содержащий адреса возврата при обработке прерываний и при выполнении вложенных подпрограмм, позволил успешно реализовать на «Сетуни 70» идею структурированного программирования Э. Дейкстры, введя операции вызова подпрограммы, вызова по условию и циклического выполнения подпрограмм. Осуществленное таким образом процедурное структурированное программирование на практике подтвердило заявленные Дейкстрой преимущества его метода: трудоемкость создания программ сократилась в 5–7 раз, благодаря исключению традиционной «отладки» тестированием на конкретных примерах, причем программы обрели надлежащую надежность, упорядоченность, понятность и модифицируемость. В дальнейшем эти особенности архитектуры «Сетуни 70» послужили основой диалоговой системы структурированного программирования ДССП, реализованной на машинах серии ДВК и на последующих персональных компьютерах [10, 11].

К сожалению, дальнейшее развитие заложенных в «Сетуни 70» возможностей путем разработки ее программного оснащения было административным порядком прекращено. Пришлось переориентироваться на компьютеризацию обучения. «Сетунь 70» стала основой для разработки и реализации автоматизированной системы обучения «Наставник» [12, 13], воплотившей принципы «Великой дидактики» Яна Амоса Коменского. Назначение компьютера в этой системе не «электронное перелистывание страниц» и не мультимедийные эффекты, а отслеживание верности понимания учащимся того, чему он учится, своевременное преодоление заблуждений и обеспечение путем обоснованно назначаемых упражнений реального овладения предметом обучения. Вместе с тем компьютер протоколирует ход занятия, предоставляя разработчику учебного материала возможность оценивать эффективность используемых дидактических приемов и совершенствовать их.

Учебный материал в «Наставнике» предоставляется учащимся в печатном виде с пронумерованными секциями, абзацами, упражнениями и справками к ошибочным ответам, благодаря чему при помощи простейшего терминала с цифровой клавиатурой и калькуляторным индикатором компьютер без гипертекстового дисплея легко и безвредно взаимодействует с обучаемым, придавая книге недостающую ей способность диалога с читателем. Создание учебных материалов для «Наставника» не связано с программированием компьютера, и, как показала практика, разработка вполне удовлетворительных пособий по математике, физике, английскому языку и другим предметам посильна школьным учителям. Дидактическая эффективность этой немудреной системы оказалась на редкость высокой. Так, курс «Базисный Фортран» студенты факультета ВМК МГУ проходили в «Наставнике» за 10–15 часов, студенты экономического факультета – за 15–20 часов, показывая затем в практикуме более совершенное умение программировать на Фортране, чем после обычного семестрового курса.

Реализованный в «Наставнике» принцип «книга-компьютер» обусловил оптимальное использование компьютера как средства обучения практически во всех отношениях: необходимая аппаратура (микрокомпьютер и подключенные к нему 3–4 десятка терминалов, подобных простейшему калькулятору) предельно дешева, надежна и легко осваивается как учащимися, так и преподавателями, работа в режиме диалога с книгой не утомительна, увлекательна и при надлежащей организации изложения гарантирует быстрое и полноценное усвоение изучаемого предмета. Применение системы в МГУ, МАИ, ВИА им. Куйбышева, в средней школе и для профессионального обучения на ЗИЛе подтвердили ее высокую эффективность в широком спектре предметов и уровней обучения. Вместе с тем «Наставник» уже более 30 лет постоянно

используется на факультете ВМиК для автоматизированного проведения контрольных работ, а также тестирования поступивших на факультет, определяющего уровень владения английским языком для комплектования однородных учебных групп.

Однако при, казалось бы, насущной потребности действенного усовершенствования процесса обучения в наш «информационный» век «Наставник» не был востребован. По-видимому, слишком прост и дешев, да и какая же это компьютерная система – без дисплея, мышки и гипертекста. Ведь ИТ-оснащенность учебного процесса все еще принято оценивать не по уровню и качеству обучения, а по количеству и мощности вовлеченных в него компьютеров.

### *Система команд машины «Сетунь»*

| Код операции | Название операции        | Содержание команды                             | Выработка $\omega$ | Время выполнения, мксек |
|--------------|--------------------------|------------------------------------------------|--------------------|-------------------------|
| -0-0         | Считывание с МБ          | $[M] \Rightarrow [\Phi_a]$                     | Сохраняется        | 7500                    |
| -000         | Ввод-вывод               | Ввод в $\Phi_a$                                | Сохраняется        | -                       |
| -0+0         | Запись на МБ             | $[\Phi_a] \Rightarrow [M]$                     | Сохраняется        | 7500                    |
| -+-0         | Нормализация             | Норм $(S) \Rightarrow (A); N \Rightarrow (S)$  | $\omega (S)$       | $275+5 N $              |
| -+00         | Сдвиг                    | Сдв $(S)$ на $(A) \Rightarrow (S)$             | $\omega (S)$       | $290+5 N $              |
| -++0         | Запись из S              | $(S) \Rightarrow (A)$                          | $\omega (S)$       | 225                     |
| 0--0         | Сложение в F             | $(A)+(F) \Rightarrow (F)$                      | $\omega (F)$       | 180                     |
| 0-00         | Посылка в F              | $(A) \Rightarrow (F)$                          | $\omega (F)$       | 180                     |
| 0-+0         | Сложение в C с посылкой  | $(A)+(C) \Rightarrow (F)$                      | $\omega (F)$       | 180                     |
| 00-0         | Запись из F              | $(F) \Rightarrow (A)$                          | $\omega (F)$       | 180                     |
| 0000         | Безусловный переход (БП) | $A \Rightarrow (C)$                            | Сохраняется        | 100                     |
| 00+0         | Запись из C              | $(C) \Rightarrow (A)$                          | Сохраняется        | 180                     |
| 0+-0         | Условный переход УП-     | $A \Rightarrow (C)$ при $\omega = -$           | Сохраняется        | 100,180                 |
| 0+00         | Условный переход УП0     | $A \Rightarrow (C)$ при $\omega = 0$           | Сохраняется        | 100,180                 |
| 0++0         | Условный переход УП+     | $A \Rightarrow (C)$ при $\omega = +$           | Сохраняется        | 100,180                 |
| +--0         | «Останов»                | «Останов»; $(A) \Rightarrow (R)$               | $\omega (R)$       | 180                     |
| + -00        | Поразрядное умножение    | $(A) \otimes (S) \Rightarrow (S)$              | $\omega (S)$       | 180                     |
| + -+0        | Посылка в R              | $(A) \Rightarrow (R)$                          | $\omega (R)$       | 180                     |
| +0-0         | Вычитание в S            | $(S) - (A) \Rightarrow (S)$                    | $\omega (S)$       | 180                     |
| +000         | Посылка в S              | $(A) \Rightarrow (S)$                          | $\omega (S)$       | 180                     |
| +0+0         | Сложение в S             | $(S) + (A) \Rightarrow (S)$                    | $\omega (S)$       | 180                     |
| ++-0         | Умножение1               | $(A) + (S) (R) \Rightarrow (S)$                | $\omega (S)$       | 320                     |
| ++00         | Умножение2               | $(S) \Rightarrow (R); (A) (R) \Rightarrow (S)$ | $\omega (S)$       | 320                     |
| +++0         | Умножение3               | $(S) + (A) (R) \Rightarrow (S)$                | $\omega (S)$       | 320                     |

**Библиографический список**

1. Кнут Д. Искусство программирования для ЭВМ. Получисленные алгоритмы. Т. 2. М.: МИР, 1977. С. 218.
2. Shannon C.E. A symmetrical notation for numbers // The American mathematical monthly. V. 57. N. 2. Feb. 1950. P. 90–93.
3. Брусенцов Н.П. Заметки о троичной цифровой технике // Архитектура и программное оснащение цифровых систем. М.: Изд-во Моск. ун-та, 1984. С. 114–123.
4. Брусенцов Н.П. Вычислительная машина «Сетунь» Московского государственного университета // Новые разработки в области вычислительной математики и вычислительной техники. Киев, 1960. С. 226–234.
5. Малая цифровая вычислительная машина «Сетунь» / Н.П. Брусенцов, С.П. Маслов, В.П. Розин, А.М. Тишулина. М.: Изд-во МГУ, 1965.
6. Жоголев Е.А. Система команд и интерпретирующая система для машины «Сетунь» // Журнал вычислительной математики и математической физики. 1961. № 3. С. 499–512.
7. Аннотированный указатель программ для вычислительной машины «Сетунь» / Н.П. Брусенцов, В.А. Морозов. М.: ОНТИ ВЦ МГУ, 1968; Вып. 2. 1971.
8. Кузнецов С.И. и др. Материалы по математическому обслуживанию ЭЦВМ «Сетунь». М.: Изд-во ВВИА им. Жуковского, 1964.
9. Брусенцов Н.П., Жоголев Е.А. Структура и алгоритм функционирования малой вычислительной машины // Вычислительная техника и вопросы кибернетики. Вып. 8. Л.: Изд-во Ленингр. ун-та, 1971. С. 34–51.
10. ДССП – диалоговая система структурированного программирования / Н.П. Брусенцов, Г.В. Златкус, И.А. Руднев // Программное оснащение микрокомпьютеров. М.: Изд-во МГУ, 1982. С.11–40.
11. Развиваемый адаптивный язык РАЯ диалоговой системы программирования ДССП / Н.П. Брусенцов, В.Б. Захаров, И.А. Руднев, С.А. Сидоров, Н.А. Чанышев. М.: Изд-во МГУ, 1987.
12. Автоматизированная система обучения «Наставник» / Н.П. Брусенцов, С.П. Маслов, Х. Рамиль Альварес // Вычислительная техника и вопросы кибернетики. Вып. 13. М.: Изд-во МГУ, 1977. С.3–17.
13. Микрокомпьютерная система обучения «Наставник» / Н.П. Брусенцов, С.П. Маслов, Х. Рамиль Альварес. М.: Наука: Физматлит, 1990.

## ИЗ ИСТОРИИ РАЗВИТИЯ И ПРИМЕНЕНИЯ КОМПЬЮТЕРНОЙ АЛГЕБРЫ В ИНСТИТУТЕ ПРИКЛАДНОЙ МАТЕМАТИКИ ИМЕНИ М.В. КЕЛДЫША

**Г.Б. Ефимов, Е.Ю. Зуева, И.Б. Щенков**

Институт прикладной математики им. М.В. Келдыша РАН

Москва, РФ

*efimov@keldysh.ru*

Дается исторический обзор и основная библиография по развитию и использованию компьютерной алгебры или символьных преобразований в Институте прикладной математики имени М.В. Келдыша Российской академии наук. Институт был создан М.В. Келдышем в 1953 году для решения важнейших комплексных научных задач, первоначально для расчетов в связи с созданием ядерного оружия. Затем это были задачи освоения космоса, кибернетика, ядерная физика и термоядерный синтез, программирование.

Институт объединил специалистов различных областей – физиков, математиков, механиков, программистов и создателей вычислительных машин. Среди них такие крупные ученые, как А.Н. Тихонов, К.И. Бабенко, И.М. Гельфанд, И.Б. Зельдович, С.П. Курдюмов, А.А. Ляпунов, А.Н. Мямлин, Д.Е. Охочимский, А.А. Самарский, К.А. Семендяев, М.Р. Шура-Бура, Т.М. Эннеев, В.С. Яблонский и многие другие. Соединение сложных задач и разнообразных математических методов с активным, пионерским использованием вычислительной техники, энтузиазмом развития ее дали немало ярких идей. Мысли о символьных вычислениях возникли вскоре после создания ЭВМ как желание научить их работать с формулами и облегчить громоздкий труд физика или механика. Компьютерная Алгебра, или Символьные Преобразования, как их называли раньше, использовалась сотрудниками института в различных областях: прикладной небесной механике, математике, робототехнике, гидромеханике, численных методах. Разрабатывались программные системы разного типа для символьных преобразований; велась работа по описанию и классификации их свойств. Был организован ряд конференций и семинаров по применению Компьютерной Алгебры в механике. В институте был разработан язык рекурсивных функций Рефал, который затем активно развивался и разнообразно использовался, в том числе для Компьютерной Алгебры. История работ, выполненных по Компьютерной Алгебре и в смежных областях, в ИПМ – одном из ведущих и самых первых компьютерных центров, и история исследований в нем естественно связана со многими исследованиями по этой теме в нашей стране. Работа поддержана РФФИ, грант 04-01-00346. По работе был издан препринт ИПМ № 27 за 2003 год.

## **КОНЦЕПЦИЯ НОВЫХ ПОКОЛЕНИЙ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ – ПОСЛЕДНЯЯ КРУПНАЯ ИНИЦИАТИВА СОЦИАЛИСТИЧЕСКИХ СТРАН В ОБЛАСТИ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И ИНФОРМАТИКИ – ВЗГЛЯД 20 ЛЕТ СПУСТЯ**

**В.Н. Захаров**

Институт проблем информатики РАН

Москва, РФ

*vzakharov@ipiran.ru*

В 1984–85 гг. под руководством директора ИПИАН Б.Н. Наумова, при активной поддержке Е.П. Велихова (тогда вице-президента АН СССР) коллектив специалистов АН СССР, академий наук республик СССР, академий наук стран Восточной Европы разработал Концепцию новых поколений вычислительных систем. В Концепции были представлены главные направления исследований и разработок, которые следовало провести, чтобы получить новые качества информационных и вычислительных систем. Выполнение Концепции предполагалось проводить в виде десяти комплексных научных проектов, осуществляемых международными исследовательскими коллективами:

- «Система обработки знаний» (ответственный исполнитель проекта – ЧССР).

Подпроекты: «Экспертные системы», «Основные средства для обработки знаний», «Системы машин обработки знаний», «Распределенные системы баз знаний», «Диалог на естественном языке», «Машина баз данных».

- «Системы обработки изображений и машинная графика» (ГДР). Подпроекты: «Создание систем цифровой обработки изображений», «Создание типовых прикладных систем автоматизированной обработки и анализа изображений».

- «Системы автоматизации проектирования вычислительных систем» (СССР). Целью проекта было создание интегрированной распределенной САПР, состоящей из системы описания и моделирования архитектурного уровня вычислительных машин, САПР СБИС и УБИС, системы автоматизации конструкторского проектирования вычислительных машин.
- «Сети ЭВМ» (ГДР). Подпроекты: «Локальные вычислительные сети», «Аппаратные и программные средства для сетевых процессоров», «Сеть интеллектуальных терминалов», «Распределенная обработка данных».
- «Системы персональных компьютеров» (СССР).
- «Отказоустойчивые системы» (СССР, ГДР).
- «Новые ВЗУ» (НРБ). Предполагалось проведение исследований в следующих направлениях: оптические ВЗУ, многослойные кремниевые дисковые ЗУ, ЗУ на ЦМД, магнитные ВЗУ, защита информации в ВЗУ.
- «Технология программного обеспечения ЭВМ новых поколений» (СССР).
- «Новые алгоритмы и архитектуры обработки информации» (СССР, НРБ).
- «Учебная информатика» (НРБ).

В 1988 году в издательстве «Akademie-Verlag Berlin» начал выходить «Журнал новых поколений вычислительных систем», в котором освещался ход выполнения работ по реализации Концепции.

В докладе предполагается проанализировать, насколько жизнь подтвердила правильность идей, заложенных в Концепцию, что удалось, а что не удалось сделать.

## ОБЕСПЕЧЕНИЕ НАДЕЖНОСТИ АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ СЕТЕВОЙ КЛАСТЕРИЗАЦИИ СЕРВЕРОВ

**В.П. Климанов, М.В. Сутягин**

МГТУ «Станкин»

Москва, РФ

*slav@stankin.ru*

Информационные системы (ИС) играют значительную роль в жизненном цикле производства продукции или оказания услуг. В настоящее время их роль и влияние на обеспечение качества продукции и услуг постоянно растет. Задачи управления эффективно решаются за счет использования средств информационных технологий. Эта тенденция будет развиваться и в дальнейшем, вызывая рост зависимости обеспечения качества продукции и услуг от средств автоматизации.

Функционирование современного промышленного предприятия связано с использованием CALS-технологий, обработкой больших объемов информации, что обеспечивается применением информационных систем.

Вместе с ростом роли ИС в обеспечении качества продукции и услуг предприятия и организации происходит рост рисков, связанных с недоступностью данных и информационных служб, необходимых для осуществления поддержки процессов жизненного цикла. Непрерывность и качество этих процессов находится в прямой зависимости от надежности информационной системы предприятия. Потери, вызванные отказом ИС, могут выражаться не только ухудшением финансовых показателей, но и приводить к уходу компании из бизнеса. Отказы ИС, связанных с обеспечением безопасности государства или сферами дея-

тельности повышенной опасности (управление пассажироперевозками, химическое производство, АЭС и т.п.), могут приводить к человеческим жертвам. Допустимое время простоя информационных служб и недоступности данных определяется исходя из характера задач, решаемых информационной системой.

Метод обеспечения надежности ИС зависит от класса готовности, который должен быть обеспечен и может варьироваться от применения надежных систем до создания резервных центров обработки информации (ЦОИ) (см. табл.).

### Классификация систем по показателю готовности

| Класс готовности | Показатель готовности, % | Тип АИС                                          | Метод обеспечения надежности                                               |
|------------------|--------------------------|--------------------------------------------------|----------------------------------------------------------------------------|
| 1                | 90                       | Обычная система (conventional)                   | Повышение надежности и качества систем                                     |
| 2                | 99                       |                                                  | Применение средств преодоления сбоев                                       |
| 3                | 99.9                     | Высокой готовности (High Availability)           | Применение избыточности (кластерные системы)                               |
| 4                | 99.99                    | Эластичная к отказам (Fault Resiliency)          | Применение избыточности на уровне систем и приложений (кластерные системы) |
| 5                | 99.999                   | Устойчивая к отказам (Fault Tolerance)           | Применение средств обеспечения непрерывного функционирования               |
| 6                | 99.9999                  | Непрерывной готовности (Continuous Availability) | Использование резервного вычислительного центра                            |

На рис. 1 представлены различные по назначению информационные системы (ИС) и их принадлежность к определенному классу готовности.

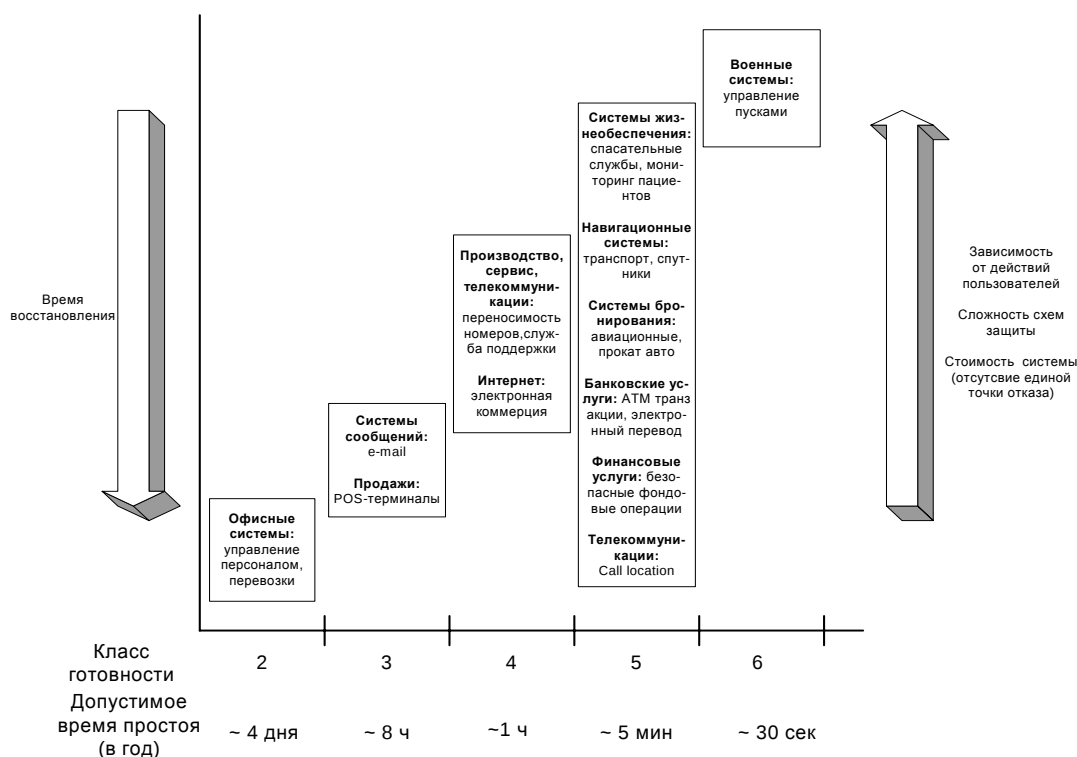


Рис. 1. Класс готовности информационных систем различного назначения

В последнее время большое внимание уделяется катастрофоустойчивым системам. Это связано с возросшей вероятностью природных и техногенных катастроф, а также террористических актов, которые могут привести к отказам, требующим большого времени для их устранения.

Катастрофоустойчивость – характеристика ИС, накладывающая дополнительные требования к программно-аппаратным средствам автоматизированных информационных систем (АИС).

Для создания катастрофоустойчивого решения требуется применение целого комплекса мер: разнесение основного и резервного ЦОИ на значительное расстояние, подготовка персонала, разработка плана действий в случае катастроф и тестирование этих планов.

Оптимальным средством для создания катастрофоустойчивых решений являются сетевые технологии. Их использование позволяет разнести основной и резервный ЦОИ на значительное расстояние и не влечет значительного увеличения стоимости системы.

Следует отметить, что при переводе системы в более высокий класс готовности ее стоимость значительно увеличивается.

В перспективе для снижения стоимости высоконадежных систем могут использоваться кластерные системы. Кластерные системы позволяют достигать требуемого уровня надежности с меньшими затратами.

В работе предлагается использование технологий, обычно применяемых для систем 3–4-го класса готовности, для создания систем 5–6-го классов и обеспечения требуемого уровня катастрофоустойчивости АИС.

Интуитивно представляется, что при использовании в таких системах сетевых кластерных технологий возможно значительное снижение стоимости системы при обеспечении требуемого уровня надежности и катастрофоустойчивости.

Кластеризация может применяться на следующих уровнях:

- аппаратного обеспечения;
- операционных систем;
- ПО промежуточного слоя;
- средств управления системами;
- приложений.

В работе основной акцент делается на кластеризацию аппаратного обеспечения.

Использование сетевых технологий позволяет обеспечить территориальное разнесение узлов кластера, что позволит обеспечить необходимый уровень катастрофоустойчивости.

Предлагается оригинальная схема сетевого кластера высокой надежности на базе двух магистралей FDDI. Технология FDDI позволяет создать катастрофоустойчивую систему за счет разнесения узлов кластера на расстояние до 100 км (рис. 2).

Подтвердить, что система относится к 5–6-му классу можно двумя способами:

- мониторингом реальной системы;
- исследованием системы на основе математической модели.

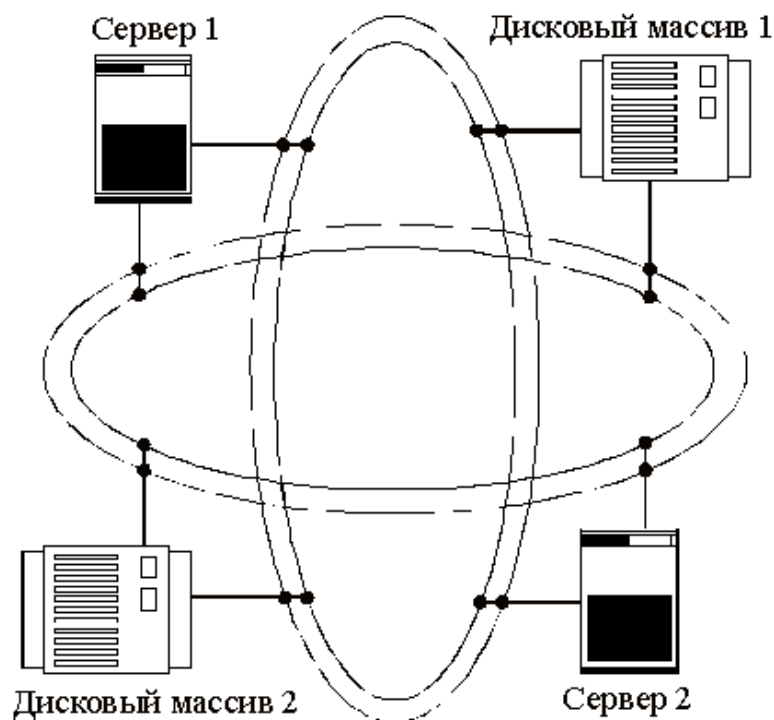
Первый способ реализовать практически невозможно – дорого и долго.

Второй способ предусматривает разработку и программную реализацию математической модели.

Модели могут быть двух видов: имитационная и аналитическая.

Главная сложность программной реализации математической модели – высокая точность моделей для систем 5–6-го класса готовности.

Рассмотрим целесообразность использования каждого из двух видов моделей.



**Рис. 2. Сетевой кластер высокой надежности  
на базе двух магистралей FDDI**

Имитационное моделирование целесообразно применять только для систем с невысоким классом готовности, т.к. для достижения точности моделирования  $10^{-7}$ , что соответствует 6-му классу готовности, необходимо обработать  $10^{10}$  отказов, что потребует большого времени вычисления. Это не позволит эффективно исследовать поведение системы в различных условиях.

Для аналитического моделирования используется аппарат марковских цепей с непрерывным временем. Это обусловлено тем, что при большом числе факторов, формирующих отказы, работают предельные теоремы теории вероятности, которые утверждают, что при одном порядке степени влияния каждого фактора на конечный результат мы имеем сходимость потока отказов к пуассоновскому закону (что подтверждено работами Погожева и Григелиониса [1, 2]). Кроме того, марковские цепи можно использовать как аппарат, определяющий верхнюю границу оценки для распределений интервалов времени безотказной работы, имеющих значения коэффициента вариации меньше и равное единице. Поэтому основным инструментом математического моделирования являются уравнения Колмогорова.

Для систем 5–6-го классов готовности и катастрофоустойчивых систем требуется учитывать особенности эксплуатации: тип контроля (непрерывный или периодический), надежность устройств переключения и контроля, стратегии переключения, число бригад восстановления и др.

Для учета этих факторов предлагается усовершенствованный критерий готовности, который учитывает особенности эксплуатации системы.

$$K = (1 - P_{\text{отк}})(1 - P_{\text{пер}})(1 - P_{\text{но}}) \prod_{i=1}^n (1 - P_{k_i}), \text{ где}$$

$P_{\text{отк}}$  — вероятность того, что система находится в состоянии отказа, обнаруживаемого контролем;

$P_{\text{пер}}$  — вероятность того, что система находится в состоянии переключения с неисправного устройства на исправное;

$P_{\text{но}}$  — вероятность того, что система находится в состоянии, когда произошел отказ, но он не обнаружен (эта вероятность характеризует особенность периодического контроля);

$P_{k_i}$  — вероятность того, что отказало  $i$ -е устройство контроля.

Критерий позволяет учитывать не только отказы основного и резервного элементов системы ( $P_{\text{отк}}$ ), но и то, что система неработоспособна в момент переключения с неисправного элемента на исправный ( $P_{\text{пер}}$ ), вероятность того, что отказ произошел, но еще не обнаружен (в случае периодического контроля  $P_{\text{но}}$ ) и ненадежность самих устройств контроля.

Для учета каждого из факторов требуется разработка отдельной аналитической модели. Например, для простого случая реализации сетевого кластера, находящегося под непрерывным контролем возникновения отказов, абсолютно надежных устройств переключения и восьми бригад восстановления граф состояний математических моделей содержит 20 элементов. А если в модели учесть фактор конечного времени переключения на резервный элемент, то число состояний увеличивается до 89.

Поэтому была разработана методика построения аналитических моделей, позволяющая учитывать особенности эксплуатации кластерной системы и существенно упростить сам процесс создания моделей [3].

Для проверки адекватности построенной аналитической модели кластера была разработана эталонная монте-карловская модель. Для реализации данной эталонной модели с заданной погрешностью  $10^{-7}$  потребовалось провести математический эксперимент, который на компьютере с процессором PIII-600 MHz длился 186 часов 2 мин 34 сек.

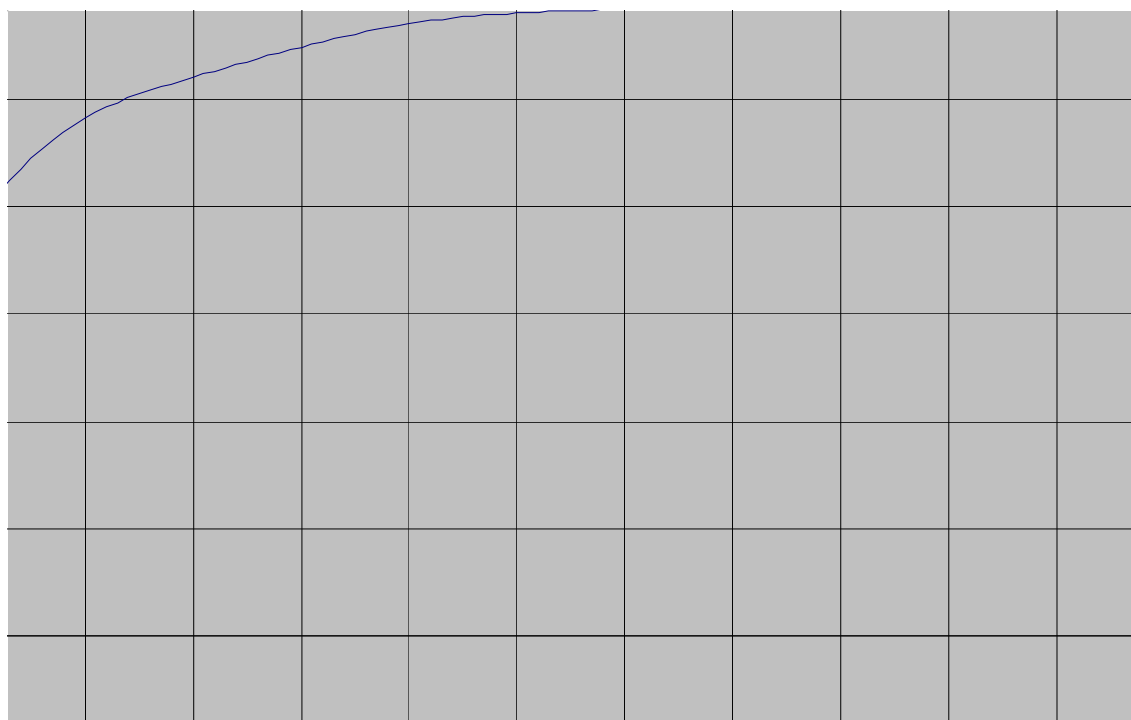
Сравнение результатов аналитического моделирования с эталонной моделью показали, что для аналитической модели выполняются условия допустимой погрешности результата –  $10^{-6}$ , а время математического моделирования на том же компьютере составляет 2–3 с. Безусловно, в этом случае аналитическая модель является предпочтительной для применения. Увеличение числа состояний системы в случае аналитического моделирования вызывает линейное увеличение времени вычисления. Для имитационного моделирования повышение точности на порядок вызывает увеличение числа отказов, которые необходимо обработать, на два порядка.

Аналитическая и монте-карловская модели были реализованы в виде программной среды моделирования надежности кластерной системы [4].

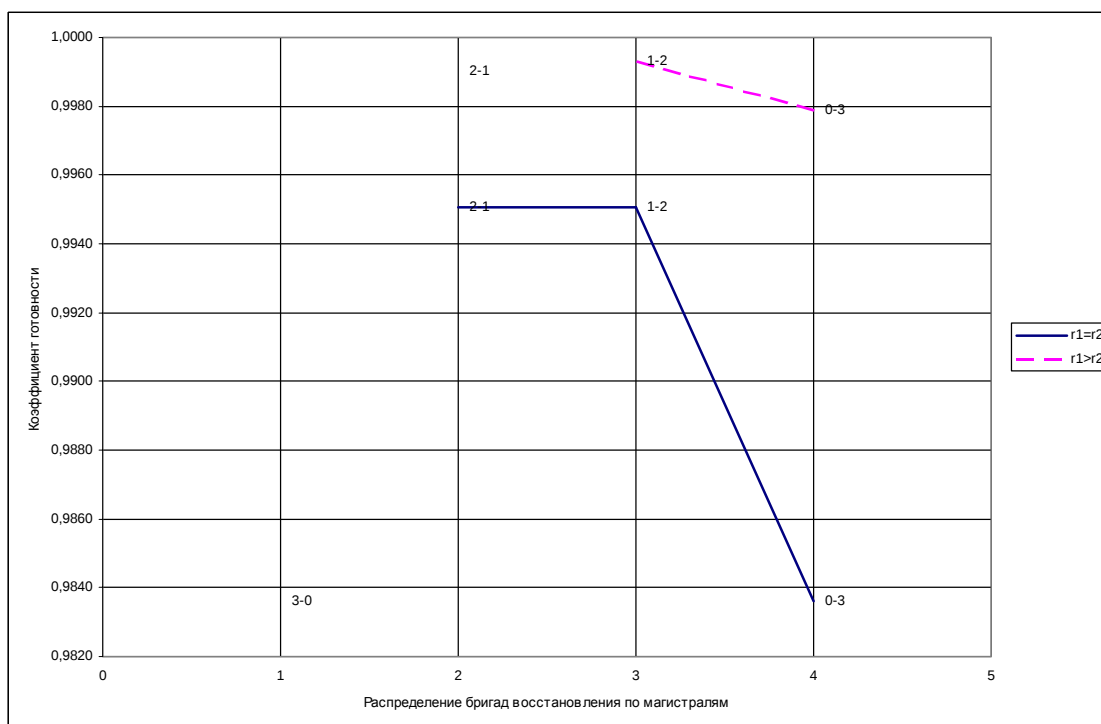
Для установившегося режима дифференциальные уравнения Колмогорова преобразуются в линейные. При решении системы линейных уравнений использовался метод Гаусса. Точность данного метода в рассматриваемом случае зависит только от округления и составляет  $10^{-9}$ – $10^{-10}$ , что достаточно для систем 6-го класса готовности.

Программная среда позволяет варьировать интенсивностями отказов и восстановлений для основной и резервной магистрали, число бригад и их распределение по магистралям, интенсивность периодического контроля, учитывать время переключения и стратегии переключения.

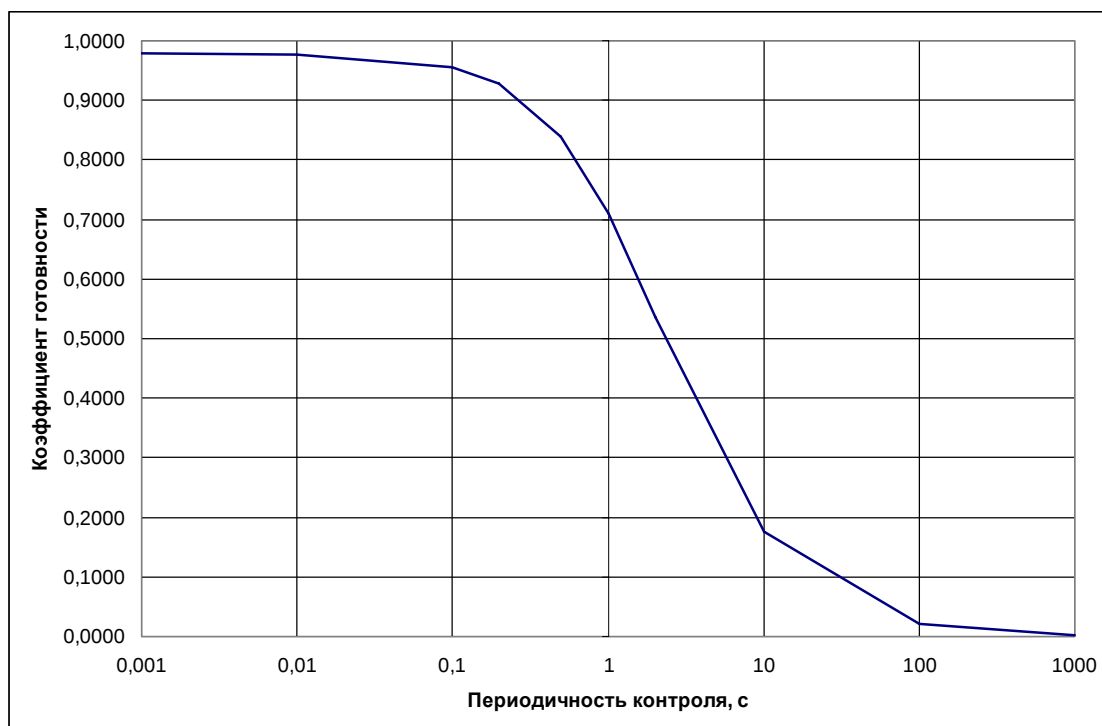
Проведено исследование зависимости готовности предлагаемой системы от условий эксплуатации (числа ремонтных бригад, их распределения по магистралям, периодичности контроля элементов системы, времени переключения). Зависимости представлены на графиках (рис. 3–6).



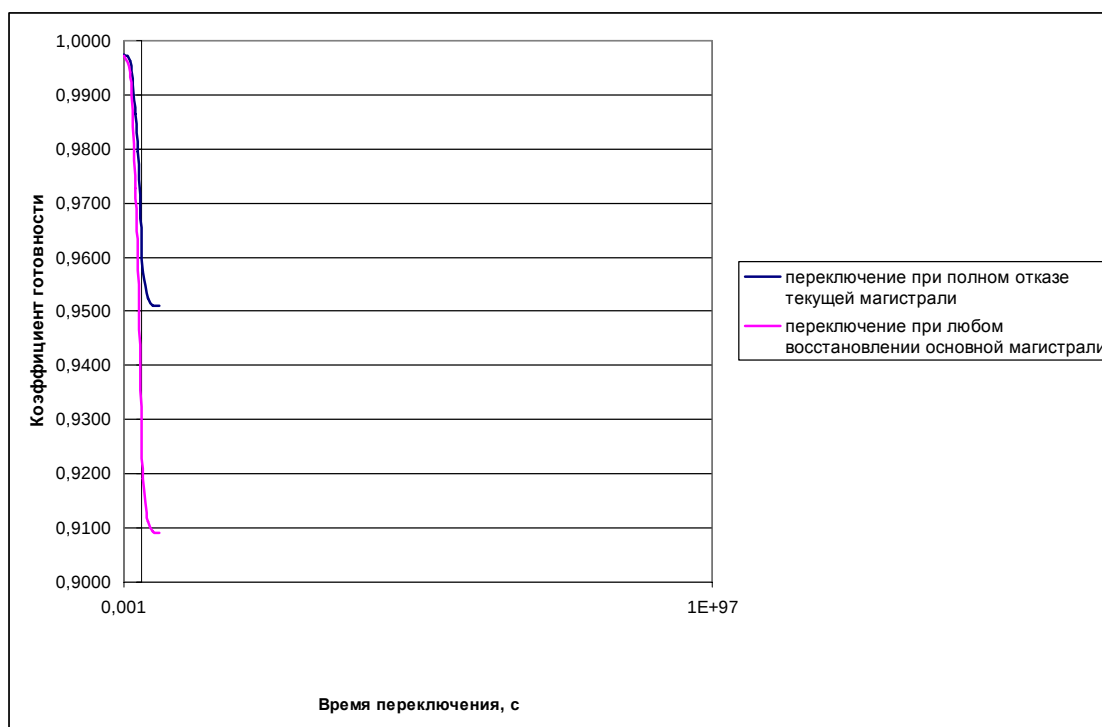
*Рис. 3. Зависимость коэффициента готовности от числа бригад восстановления*



*Рис. 4. Зависимость коэффициента готовности от распределения бригад восстановления и загрузки магистралей*

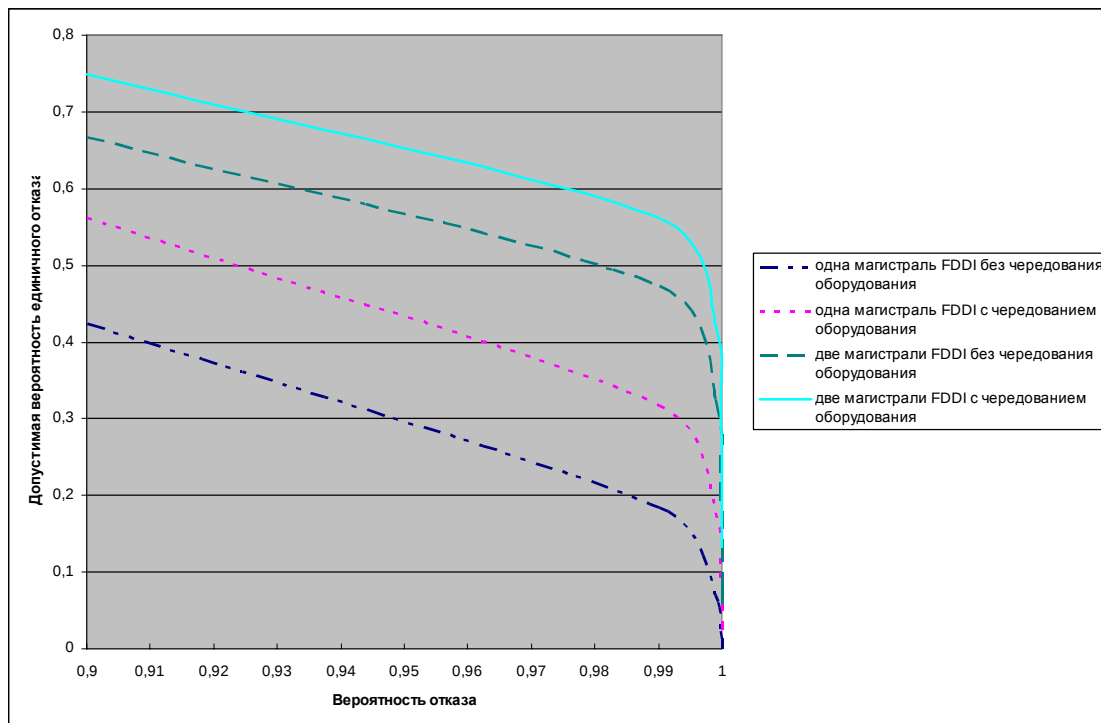


**Рис. 5. Зависимость коэффициента готовности от периодичности контроля резервной магистрали**



**Рис. 6. Зависимость коэффициента готовности от времени и стратегии переключения**

Проведено сравнение предлагаемой схемы организации кластера с типовой по допустимой вероятности единичного отказа при заданной готовности системы (рис. 7). Выявлена зависимость коэффициента готовности от способа размещения оборудования.



**Рис. 7. Допустимые вероятности единичного отказа для различных показателей готовности**

Таким образом, в данной статье рассматривались вопросы повышения надежности АИС за счет сетевой кластеризации.

Представлена методика построения аналитических моделей и программная среда моделирования, которые позволяют значительно сократить время проектирования ИС и детально изучить влияние технологических особенностей эксплуатации проектируемой ИС и выбрать наиболее рациональное решение с точки зрения критерия цена/качество.

Представленные математические модели надежности и критерий, учитывающий особенности эксплуатации системы, являются универсальными и могут быть использованы для анализа надежности других технических систем.

### **Библиографический список**

1. Погожев И.Б. Оценка отклонений потока отказов в аппаратуре многократного использования от пуассоновского потока // Кибернетику – на службу коммунизму. Т. 2. М.: Энергия, 1964. С. 228–245.
2. Григелионис Б.И. Предельные теоремы для сумм процессов восстановления // Кибернетику – на службу коммунизму. Т. 2. М.: Энергия, 1964. С. 246–265.
3. Климанов В.П., Сутягин М.В., Родионов А.Э., Ермаков А.А. Модели оценки надежности кластерной системы специального назначения // Международный форум информатизации – 2002: Докл. междунар. конф. «Информационные средства и технологии». 14–16 октября 2003 г. Т. 3. М.: Янус-К, 2003. С. 123–131.

4. Сутягин М.В. Обеспечение надежности автоматизированных информационных систем на основе сетевой кластеризации серверов: Дис. ... канд. техн. наук. М., 2004. 185 с.

## ВЛИЯНИЕ ПРОГРАММИРОВАНИЯ НА АРХИТЕКТУРУ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ (ОТ ПРОШЛОГО К БУДУЩЕМУ)

**Л.Н. Королев**

Московский государственный университет (факультет вычислительной математики и кибернетики)

Москва, РФ

*krh@cs.msu.ru*

Если раньше большее влияние на архитектуру оказывали требования разработчиков системного программного обеспечения, то в новых разработках главный упор сделан на достижения высокой эффективности микрораспараллеливания с учетом суперскалярности процессоров.

Основная тяжесть по «добыче» эффективности возлагается на разработчиков оптимизирующих трансляторов. Бытовавший ранее лозунг приближения архитектуры машин к языкам высокого уровня, по-видимому, в коммерческих разработках отброшен окончательно. Однако некоторые следы такого подхода можно отметить в микропроцессоре E2k, разработанном под руководством Б.А. Бабаяна.

Последние десятилетия потребовали от разработчиков учета требований обработки мультимедийных данных, что увеличило число типов данных, поддерживаемых аппаратурой.

Важными представляются работы, выполнявшиеся коллективом под руководством академика В.С. Бурцева и продолжающиеся в настоящее время, по созданию «Ассоциативной машины». Этот многообещающий подход, позволяющий реально осуществить принцип работы машин, управляемых потоком данных: «выполняй действие сразу в тот момент, когда будут готовы операнды», как представляется, позволит достичь максимальной эффективности динамического распараллеливания.

Несомненный интерес представляют работы, выполненные в свое время в ИТМ и ВТ по разработке архитектуры модульного конвейерного процессора – МКП (А.А. Соколов, А.Ю. Бяков и др.). Если следовать современной терминологии, это была двудерная суперскалярная архитектура процессора, включавшая блок параллельно работающих функциональных устройств, с возможностью реализации режима SIMD (векторный блок).

Что касается отражения в архитектуре вычислительных систем требований распределенных вычислений, то несомненную значимость имеют разработки, связанные с созданием АС-6, выполненные под руководством академика В.А. Мельникова. Эти работы, выполненные в прошлом столетии, явились предтечей того, что сейчас делается в мультипроцессорных системах – выделяются два типа связей: одна наиболее быстрая для обмена информацией между процессорными узлами, другая более медленная для связи с «далекими» абонентами.

Эти примеры подтверждают известную истину, что всякое новое есть хорошо забытое старое. Желательно только, чтобы «забытие» было бы не нарочитым!

Экскурс в историю и сегодняшний день развития архитектур процессоров говорит о том, что «голубая мечта» создателей программного обеспечения о платформенной независимости их разработок, о программной преемственности компьютеров всех фирм и народов, о появлении некоего стандарта архитектуры, которого придерживались бы все разработчики, так и останется неосуществимой.

Можно только уповать на законы естественного отбора, но жаль, что такой отбор требует очень продолжительного времени, особенно когда «среда обитания» (технический прогресс в элементной базе) непрерывно и непредсказуемо меняется.

## РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ АВТОМАТИЗАЦИИ АЭРОДИНАМИЧЕСКИХ ИССЛЕДОВАНИЙ

**В.С. Криворученко**

ФГУП «Центральный аэрогидродинамический институт им. профессора Н.Е. Жуковского»

Жуковский, РФ

*kvs\_kvs@mail.ru*

Центральный аэрогидродинамический институт им. проф. Н.Е. Жуковского (ЦАГИ) – головной институт авиационной отрасли России. С середины 60-х в нем были развернуты работы по созданию первой централизованной вычислительной системы (ЦВС-1) для комплексной автоматизации экспериментальных исследований в аэродинамических трубах (АДТ). В качестве ядра системы использовались две ЭВМ ДНЕПР-1 с дополнительными периферийными устройствами. К ЭВМ по линиям связи подключалась аппаратура, которая осуществляла сбор, преобразование и передачу данных.

Следующий вариант системы – ЦВС-2 представлял собой иерархическую звездообразную структуру, состоящую из двух уровней. На первом уровне были установлены измерительно-вычислительные комплексы (ИВК), оснащенные МИНИ-ЭВМ типа PDP и Mitra, устройствами связи с измерительной и управляющей аппаратурой АДТ. На второй уровень – Центральный Блок на базе трех ЕС ЭВМ: возлагались задачи накопления и архивации данных, полученных с ЭВМ 1-го уровня, их комплексной обработки, погружения в Банк Данных и последующей передачи данных в организации отрасли.

Диалоговое и программное взаимодействие в ЦВС-2 основано на обмене файлами и вызове процедур через центральный интерпретатор системы со специализированной БД, который по современным понятиям представляет собой портал. На прикладном уровне программное обеспечение связи было «прозрачно» для программ пользователей.

Коммуникации в системе поддерживали программируемые контроллеры на базе микро-ЭВМ «Электроника-60». Они подсоединялись непосредственно к мультиплексному каналу ЕС ЭВМ и за счет программной логики эмулировали интерфейсы и протоколы связи. Это позволило объединить в единую коммуникационную среду десятки ЭВМ и устройств, а через глобальные сети – и удаленные.

По техническим характеристикам система ЦВС-2 в конце 80-х вполне соответствовала мировому уровню и обеспечивала требуемый уровень информационного сопровождения проектов по созданию летательных аппаратов. Системный подход к построению единого информационного пространства, формализация описания данных на основе единого метаязыка, единая терминология, использование стандартизованных электронных документов, транспортабельное программное обеспечение для разных типов ЭВМ и ОС – по современным понятиям это и есть CALS-технологии.

Созданные в ЦАГИ распределенные системы автоматизации аэродинамических исследований постоянно модифицируются и расширяют свои возможности, оперативно отслеживая текущие и перспективные потребности аэрокосмической индустрии государства.

## ИСТОРИЯ СОЗДАНИЯ МНОГОПРОЦЕССОРНЫХ СИСТЕМ С ПРОГРАММИРУЕМОЙ АРХИТЕКТУРОЙ

**И.И. Левин**

НИИ многопроцессорных вычислительных систем им. академика А.В. Каляева

Таганрог, РФ

*levin@ms.tsure.ru*

В последней четверти XX века в НИИ многопроцессорных вычислительных систем Таганрогского государственного радиотехнического университета (НИИ МВС ТГРУ) под руководством академика РАН А.В. Каляева была разработана, теоретически и экспериментально подтверждена концепция многопроцессорных систем с программируемой архитектурой [3]. В процессе конструирования архитектура таких систем не формируется окончательно и остается в определенном смысле незавершенной и открытой. Аппаратно-программные свойства МВС предоставляют пользователю возможность как на этапе трансляции задачи, так и в процессе ее решения программировать и настраивать архитектуру системы на виртуальные вычислители, адекватные информационной структуре решаемой задачи или ее фрагмента.

Несмотря на то что концепция МВС с программируемой архитектурой была разработана в начале 80-х годов прошлого века, реальные образцы универсальной МВС ПА были созданы только в конце 90-х годов. Основной причиной этого является необходимость создания специальной элементной базы. Такая элементная база – макропроцессорный комплект СБИС, разработанный в НИИ МВС ТГРУ, была создана в СКБ «Белмикросистем» НПО «Интеграл» (г. Минск) в 1992 – 1994 годах.

В 1996 году на основе оригинальной макропроцессорной элементной базы был создан базовый модуль универсальной МВС с программируемой архитектурой. На одной плате было реализовано 32 элементарных процессора (2 макропроцессора), связанных между собой по полному графу.

Появившиеся ПЛИС позволили на практике реализовать теоретические идеи на качественно новом уровне. На основе указанных принципов разработан и создан ряд базовых модулей для построения многопроцессорных систем. Модульно-наращиваемая многопроцессорная вычислительная система с двухуровневой программируемой архитектурой, содержащая 4 базовых модуля, имеет производительность 200 Гфлопс в объеме 40 дм<sup>3</sup>.

Результаты теоретических исследований и практического использования созданных МВС показывают, что многопроцессорные вычислительные системы с программируемой архитектурой обеспечивают реальную производительность, которая близка к пиковой производительности на различных классах решаемых задач, что недостижимо для МВС с традиционной архитектурой. Это открывает широкие перспективы для создания парка высокопроизводительных вычислителей, начиная от ускорителей к персональным ЭВМ и заканчивая построением сверхвысокопроизводительных вычислительных комплексов, которые по характеристикам «реальная производительность/стоимость» в несколько раз превосходят лучшие зарубежные образцы суперкомпьютеров.

## ФУНКЦИОНАЛЬНО-ОРИЕНТИРОВАННЫЕ ПРОЦЕССОРЫ ДЛЯ СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

**Н.А. Лукин**

Институт машиноведения УрО РАН

Екатеринбург, РФ

*nicklookin@mail.ru*

Совершенствование систем реального времени приводит к децентрализации вычислений и, как следствие, к появлению в их составе функционально-ориентированных процессоров (ФОП). Создание рациио-

нальных или оптимальных архитектур ФОП возможно лишь на основе совместной оптимизации алгоритмов и архитектур. Особенно это актуально для варианта реализации ФОП как «Системы на кристалле». На примере задачи вычисления математических функций с гарантированной точностью раскрываются все этапы архитектурного синтеза – от выбора численного метода до формирования архитектуры СБИС ФОП.

## ЗЕЛЕНГРАДСКИЙ ЦЕНТР МИКРОЭЛЕКТРОНИКИ – ИСТОРИЯ СОЗДАНИЯ

**Б.М. Малашевич**

ОАО «Ангстрем»

Москва, РФ

*mbm@angstrom.ru*

В 1958 г. главный инженер КБ-1 Ф.В. Лукин поручает А.А. Колосову изучение проблем микроэлектроники. А.И. Шокин начинает создание новой подотрасли «Микроэлектроника», посылает специалистов на стажировку в США. В 1959 г. НИИ-35 (НИИ «Пульсар») начал работы по германиевым ИС.

В 1960 г. в КБ-1 выходит монография А.А. Колосова «Вопросы микроэлектроники», создается лаборатория микроэлектроники. Через год в НИИ-35 создается отдел микроэлектроники.

В 1961 г. с результатами работ А.А. Колосов обращается к К.И. Мартюшову, в Ленинграде проводится конференция по микроэлектронике, доклад А.А. Колосова, совещание у А.И. Шокина (А.А. Колосов, К.И. Мартюшов), решение о создании Центра микроэлектроники (ЦМ). Начало подготовки Постановления о ЦМ. Штаб в НИИ-35.

1962 г. Разговор А.И. Шокина с Н.С. Хрушевым, выставка с докладом в Президиуме ЦК КПСС. Рассмотрение архитектурных проектов, Н.С. Хрушев увязывает будущий Зеленоград с микроэлектроникой. А.И. Шокин организует визит Н.С. Хрущева в КБ Ф.Г. Староса, доклад Н.С. Хрущеву проекта Постановления. Выход Постановления об образовании ЦМ. Образование: НИИМП с Компонентом, НИИТМ с Элионом.

1963 г. Назначение Ф.В. Лукина директором ЦМ. Образование: НИИТТТ с Ангстремом, НИИМВ с Элмой. В НИИТМ созданы вакуумные напылительные установки. В НИИМП разработано первое в стране изделие микроэлектроники – радиоприемник «Микро».

1964 г. Образование: НИИМЭ с Микроном, НИИФП. Ангстрем серийно производит радиоприемник «Микро». В НИИТТ разработаны и изготовлены первые в стране ИС серии «Тропа».

1965 г. Образование МИЭТ с Протоном (1972). Ангстрем серийно производит ИС «Тропа». Микроном выпущены первые в Зеленограде полупроводниковые ИС «Иртыш».

1966 г. Элма выпускает 15 видов разработанных НИИМВ материалов, Элион – 20 типов созданного НИИТМ оборудования.

1968 г. Образование ЦБПИМС. НЦ сдан заказчику проект и образец суперЭВМ «Алмаз».

1969 г. Образование СВЦ с Логикой (1975). НЦ выпускает более 200 типов ИС. ИС «Тропа» первыми в мире облетели Луну и вернулись на Землю.

1971 г. В НЦ работает 12800 человек. В СВЦ разработана и испытана суперЭВМ «5Э53», документация передана на ЗЭМЗ.

1975 г. Число разработанных в НЦ ИС достигло 1020 типов.

1976 г. Создано НПО «Научный Центр» – 39 предприятий в разных городах, 80 тыс. человек.

### Модулярные суперЭВМ

В 1955 г. чехи М. Валах и А. Свобода предложили новую непозиционную систему счисления – «Систему остаточных классов» (СОК), арифметику в СОК назвали модулярной. Первым в СССР СОК оценил Ф.В. Лукин (КБ-1) и обратил на нее внимание И.Я. Акушского и Д.И. Юдицкого (СКБ-245). Эта краткая информация дала старт исследованиям И.Я. Акушского и Д.И. Юдицкого.

В 1960 г. Ф.В. Лукин (директор НИИ-37 (НИИ ДАР)) пригласил Д.И. Юдицкого и И.Я. Акушского для разработки ЭВМ для РЛС. В 1963 г. их коллективом разработана модулярная 45-разрядная ЭВМ Т-340А, работавшая в полигонной РЛС. Вариант модулярной арифметики и архитектура Т-340А разработаны И.Я. Акушским, Д.И. Юдицким и Е.С. Андриановым. В 1966 г. разработана ЭВМ К-340А – базовая для всех РЛС НИИ-37, выпущено более 50 ЭВМ. Быстродействие К-340А 2,4 млн. оп./с.

В 1963 г. Ф.В.Лукин назначен директором создаваемого зеленоградского Центра микроэлектроники и приглашает коллектив Д.И. Юдицкого для конкурсной разработки супер-ЭВМ «Алмаз» по заказу генерального конструктора ПРО Г.В. Кисунько. В 1967 г. образец 45-разрядной ЭВМ разработан, изготовлен и принят заказчиком. Быстродействие Алмаза до 30 млн. оп./с. Конкурс выигран, получен заказ на ЭВМ 5Э53 с уточненными характеристиками.

В начале 1971 г. 40-разрядная ЭВМ 5Э53 с быстродействием 40 млн. оп./с. разработана, образец изготовлен и испытан, комплект документации передан на освоение Загорскому электромеханическому заводу. В 1972 г., когда завод уже приступил к изготовлению ЭВМ, работы по развитию ПРО К.В. Кисунько были прекращены, а с ними и работы по производству 5Э53.

### *Зеленоградские МП и ЭВМ с архитектурой «Электроника НЦ»*

В 1973 г. СВЦ разработана архитектура 16-разрядных магистрально-модульных мини-ЭВМ «Электроника НЦ» и мини-ЭВМ «Электроника НЦ-1» на ее основе. Производилась Псковским ЗРД. В 1974–75 гг. СВЦ на НЦ-1 разработан Центр коммутации сообщений «Юрюзань», установленный в аэропорту Пулково. После ликвидации СВЦ Минэлектронпром отказался тиражировать Юрюзань. В 1974–75 гг. СВЦ разработан комплекс вычислительных средств «Связь-1». Производился заводом «Красная заря».

В 1974–75 гг. СВЦ (откорректированная архитектура НЦ, схемотехника) и НИИТТ (топология и технология) разработан МПК серий К(Н)587. Позже по этой архитектуре были разработаны МПК серий К588 (с ПО «Интеграл»), К1883 (с ф. «Роботрон»), К1802 (с НИИМЭ) и однокристалльная микроЭВМ К1801ВЕ1 (НИИТТ).

В 1974–75 гг. на К587 разработаны одноплатная микро-ЭВМ «Электроника НЦ-01», двухплатная «Электроника НЦ-02» и блочная «Электроника НЦ-02М». Выпускались небольшими тиражами.

По результатам откорректирована архитектура НЦ и сформирован ряд микроЭВМ «Электроника НЦ-03, -04, -05».

В 1975–76 гг. на К587 разработаны микроЭВМ «Электроника НЦ-03Т, НЦ-03Д, НЦ-03С, НЦ-04Т», выпускались сотнями шт.

В 1977 г. на К587 для космических объектов разработан и поставлялся бортовой компьютер «Электроника НЦ-04У» на керамических платах.

В 1977–78 гг. на К587 разработан и выпущена пария портативного медицинского комплекса «Электроника Тонус НЦ-01», а на К588 – СЧПУ «Электроника НЦ-31», выпускалась несколькими заводами в больших объемах.

В 1980 г. на НЦ-03Т, НЦ-04Т разработан и серийно производился многофункциональный концентратор телеграфных каналов «Электроника НЦ-32».

В 1981 г. разработана микроЭВМ «Электроника НЦ-05Т», изготовлено пять штук.

В 1981–82 гг. на К1801ВЕ1 разработаны и изготовлены одноплатная микроЭВМ «Электроника НЦ-8001», микро-ЭВМ переменного состава «Электроника НЦ-8020» и первый в стране персональный компьютер «Электроника НЦ-8010».

В 1981 г. приказом Минэлектронпрома архитектура НЦ запрещена, заменена более старой архитектурой PDP-11. Серийное производство НЦ-03Т, -03Д, -04Т, -31, -32 продолжалось. Работы над К1801ВЕ1, НЦ-05Т, -8001, -8010, -8020 (лучшими отечественными достижениями) прекращены.

### *Зеленоградские DEC-совместимые МП и ЭВМ*

Запрет министром архитектуры «НЦ» принудил зеленоградцев разрабатывать DEC-совместимые микропроцессоры и микро-ЭВМ, но с собственными структурными и схемными решениями и более высокой (в разы) интеграцией.

В 1981–84 гг. разработаны 16-разрядные n-МОП микропроцессоры К1801ВМ1 (LSI-11, «Электроника 60»), К1801ВМ2 (LSI-11/2, «Электроника 60М») и К1801ВМ3/4 (LSI-11/23, «Электроника 60-1»). Для микропроцессоров на БМК К1801ВП1 НЦ (К1801ВП1-xxx) и ЛОЭП «Светлана» (К1809xx) создали более 30 однокристалльных периферийных микроконтроллеров. Все БИС со встроенной шиной МПИ (Q(22)-Bus).

Через 2–3 года разработаны КМОП варианты микропроцессоров 1806ВМ2, 1806ВМ3/4 и микроконтроллеров и 32-разрядный комплект Л1839 (VAX-11/750).

В 1981–86 гг. на К1801 разработаны одноплатные микроЭВМ «Электроника НЦ-8001Д» (К1801ВМ1), НЦ-8001ДМ (К1801ВМ2), НЦ-8001ДА (К1801ВМ3), НЦ-8001ДС (К1801ВМ3/4). Выпускались сотнями тысяч.

В 1981–86 гг. на НЦ-8001Дх построены ДВК-1 (Электроника НЦ-8020/1), ДВК-2 (НЦ-8020/2), ДВК-3 (НЦ-8020/3) и ДВК-4 (НЦ-8020/4). Львовским ПО «Полярн» разработан IBM PC-совместимый процессор МС1686 для ДВК (плата) – адаптер на ДВК ОС ПК ЕС 1840. ДВК выпускались десятками тысяч.

На основе К1801ВМ1 и К1801ВМ2 построены компактные ПК (моноблок с клавиатурой) «Электроника НЦ-8010Д», «Электроника БК-0010(Ш), БК-0011(Ш)» и «Электроника УК-НЦ». Выпускались шестью заводами десятками тысяч.

На основе ДВК, БК-0010(Ш), БК-0011(Ш) и УК-НЦ выпускались комплекты учебной вычислительной техники (КУВТ). Из шести типов КУВТ, принятых в стране для компьютеризации школ, три зеленоградского происхождения: КУВТ-ДВК (ДВК-2МШ учителя + 12 ДВК-1 учеников), КУВТ-86 (ДВК-2МШ + 12 БК-0010Ш/БК0011Ш) и КУВТ-УКНЦ (УК-НЦ + 12 УК-НЦ). КУВТ выпускались восемью заводами, было поставлено около 270000 ЭВМ.

На микропроцессорном ядре 1806ВМ2 выпущен карманный Basic-микрокомпьютер «Электроника МК-85». В 1985–2002 гг. их поставлено около 150000.

На 1806ВМ2, 1806ВМ3/4 разработаны космические бортовые компьютеры: «Электроника НЦ-40Б» для орбитальной космической станции «Мир» и других спутников, многоцелевой бортовой вычислительный комплекс «Нептун», унифицированные встраиваемые микроЭВМ «Электроника НЦ-84Б» (1806ВМ2) и «Электроника НЦ-96Б» (1806ВМ3/4) на керамической плате 100 x 110 мм, монтируемые на плате потребителя как БИС.

На МПК Л1839 разработаны инженерная рабочая станция ИРС-32 с развитой периферией для САПР и адаптер VAX – IBM PC в виде слота VAX/PC с шиной ISA.

## ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В УКРАИНЕ. СОСТОЯНИЕ И ПЕРСПЕКТИВЫ

**Л.Б. Малиновский**

Институт кибернетики им. В.М. Глушкова НАН Украины

Киев, Украина

*icfest@icfest.kiev.ua*

Дан обзор современного состояния ИТ бизнеса в Украине в области телекоммуникаций, компьютерных технологий, вычислительных технологий, а также работ по оказанию услуг в сфере программного обеспечения и консалтинговых услуг, интеграции и оптимизации с применением информационных технологий в реальном бизнесе.

Приведена информация о ведущих украинских коммерческих ИТ-компаниях, которые, на текущий момент, по производительности труда и применяемым технологиям управления процессами в ИТ-индустрии вполне отвечают мировому уровню. Состояние дел в области образования и подготовки ИТ-специалистов и перспективы создания конкурентоспособного на мировом рынке программного обеспечения.

## РАЗВИТИЕ АРХИТЕКТУРЫ ЭВМ ОТ БЭСМ-6 ДО СУПЕРКОМПЬЮТЕРОВ

**Ю.И. Митропольский**

Институт системного анализа РАН

Москва, РФ

*mitr@imms.ru*

Рассмотрены архитектура, основные принципы и особенности вычислительной машины БЭСМ-6 и системы обработки данных АС-6, созданных под руководством академика С.А. Лебедева в коллективе лаборатории № 1 Института точной механики и вычислительной техники АН СССР, которой руководил академик В.А. Мельников. Рассмотрены работы по созданию суперкомпьютерной системы «Электроника СС БИС-1», выполненные под руководством академика В.А. Мельникова. Изложены основные результаты проводимых в настоящее время исследований по мультиархитектурным (неоднородным) вычислительным суперсистемам.

Вехой в отечественной вычислительной технике стала ЭВМ БЭСМ-6, главным конструктором которой был академик С.А. Лебедев, а его заместителем академик В.А. Мельников. Машина БЭСМ-6 предназначалась для решения крупных научно-технических задач, что, естественно, отразилось как на ее архитектуре, так и на выборе системы элементов и конструкции.

В начале 60-х годов отечественной промышленностью были созданы высокочастотные транзисторы и диоды, на основе которых была разработана элементная база машины. В ее состав входили диодные логические схемы и усилители на переключателях тока с подвешенным источником питания. Важными особенностями системы элементов являлись высокая скорость переключения и очень высокая нагрузочная способность как по входу, так и по выходу. Оригинальная конструкция машины обеспечила высокую плотность размещения блоков и уменьшение длины межблочных связей. В результате была достигнута высокая тактовая частота – 10 МГц.

Система синхронизации обеспечивала возможность функционирования конвейера на тактовой частоте, что использовано в большинстве схем, в частности в арифметическом устройстве и в устройстве управления. На следующем уровне темп конвейера определялся циклом работы буферной памяти, который равен трем тактам. Для согласования пропускных способностей процессора и оперативной памяти применялось расслоение оперативной памяти и неадресуемая буферная память с ассоциативным поиском. По существу это была кэш-память, реализованная на 10 лет раньше, чем в модели 85 системы IBM/360, однако по технологическим причинам объем этой памяти был невелик. К особенностям системы команд следует отнести одноадресную структуру команды, наличие операций с плавающей запятой, индекс-регистров для модификации адресов, двух форматов команд в зависимости от длины адреса.

Важной особенностью машины явились аппаратные и программные средства для обеспечения мультипрограммного режима. К ним относятся виртуальная адресация памяти со страничной организацией, система прерывания, наличие нескольких режимов выполнения команд в процессоре и соответствующие программы операционной системы. При реализации подсистемы ввода-вывода ставилась задача обеспечения высокой пропускной способности при обмене с устройствами памяти на внешних магнитных носителях и обслуживания достаточного числа электромеханических устройств ввода и вывода. В машине было реализовано 7 быстрых направлений обмена (в современных терминах – 7 селекторных каналов) и набор медленных направлений, аппаратура для которых ограничивалась минимальным набором согласующих элементов и схем связи этих элементов с процессором. Функционирование медленных направлений (образующих мультиплексный канал) обеспечивалось программами работы с каждым конкретным типом устройства.

При разработке БЭСМ-6 была создана оригинальная система представления схемной документации и связанная с ней методология проектирования [1]. Она была основана на формульном описании логических схем и системы бланков, в которых содержалась информация о логической схеме блока и адреса его соединений с другими блоками.

Указанные выше аппаратные средства обеспечили создание многопользовательской операционной системы. За время эксплуатации машины было разработано несколько вариантов операционных систем, а также трансляторы с автокода и распространенных языков высокого уровня.

ЭВМ БЭСМ-6 выпускалась начиная с 1967 г. и установила рекорды по продолжительности выпуска – более 15 лет – и по продолжительности эксплуатации – более 20 лет. Но ее влияние на развитие отечественной вычислительной техники определялось не столько длительностью эксплуатации, а тем, что заложенные при создании машины идеи оказались весьма плодотворными. Несколько поколений инженеров и программистов, работавших на машине, были воспитаны на этих идеях.

Следующей крупной разработкой, выполненной под руководством главного конструктора В.А. Мельникова, была система обработки данных АС-6. Установка и эксплуатация БЭСМ-6 в вычислительных центрах, где выполнялась обработка больших объемов данных, поступающих от большого числа абонентов, в частности в Центре управления полетами, послужила стимулом создания системы АС-6. В этих центрах узким местом являлось небольшое число внешних устройств и низкая пропускная способность подсистемы ввода-вывода БЭСМ-6. Разработка началась с создания аппаратуры сопряжения для БЭСМ-6 (отсюда и название АС-6). На первом этапе ставились задачи стыковки БЭСМ-6 с АС-6, которая должна была обеспечить подключение большого числа телеграфных и телефонных каналов, каналов приема телеметрической информации, а также увеличение объема памяти на магнитных дисках и существенное увеличение числа периферийных устройств. Однако по мере накопления опыта по использованию оборудования первого этапа

стало очевидно, что в системе необходимы более мощные средства для обработки данных и, главное, необходимо наличие возможности наращивания системы за счет подключения дополнительных машин и устройств. Все эти обстоятельства привели к постановке задачи разработки многомашинной системы с развитыми средствами реконфигурации.

В основу реализации системы легли идеи специализации подсистем и устройств и унификации в рамках системы каналов обмена.

Кроме БЭСМ-6 в систему входили центральный процессор АС-6, периферийная машина ПМ-6, дополнительные устройства оперативной памяти, контроллеры магнитных дисков, контроллер приема телеметрической информации. Все эти устройства объединялись в систему в качестве абонентов канала 1-го уровня.

Система АС-6 с 1973 г. находилась в опытной эксплуатации, при этом продолжались работы по ее развитию. В 1975 г. она использовалась при проведении работ по программе совместного советско-американского проекта «Апполон – Союз». Сдача системы в полном объеме была проведена в 1979 г.

В системе АС-6 были впервые реализованы новые идеи, явившиеся основой разработок суперЭВМ и фундаментальных исследований по архитектуре перспективных вычислительных систем. Прежде всего, необходимо отметить следующие особенности:

- АС-6 – это неоднородная многомашинная вычислительная система;
- проблемная ориентация ЦП АС-6 на решение задач по управлению сложными объектами и эффективную трансляцию;
- функциональная специализация периферийной машины ПМ-6 и других вспомогательных устройств;
- специализация внутрисистемных каналов.

По мере создания и эксплуатации системы стало очевидно несоответствие новых архитектурных идей и возможностей элементной базы. С целью дальнейшего развития этого направления в 1973 г. был разработан проект системы БЭСМ-10, в котором на основе задела, полученного при создании АС-6, и использования высокоскоростных интегральных схем типа ЭСЛ планировалось создание перспективной вычислительной системы. Однако этот проект не был поддержан Министерством радиопромышленности СССР.

Продолжение работ в этом направлении было осуществлено под руководством академика В.А. Мельникова в рамках создания суперкомпьютерной системы «Электроника СС БИС». Разработка суперкомпьютерной системы «Электроника СС БИС-1» базировалась на том научном багаже, который был накоплен при создании БЭСМ-6 и АС-6. Однако для достижения производительности на два порядка величины большей, чем в этих машинах, было необходимо освоение нового технологического уровня и разработки соответствующей ему архитектуры [2].

В первоначальном проекте системы рассматривались возможности включения в ее состав следующих проблемно-ориентированных подсистем: основная машина с векторно-конвейерным процессором; матричная машина; машина для логической обработки данных. Кроме того, рассматривались возможности включения следующих функционально-специализированных подсистем: периферийная машина; контроллер внешней полупроводниковой памяти; контроллер дисковой памяти; внешние машины; управляющие машины. С учетом имевшихся ресурсов и первоочередных задач было принято решение отложить разработку матричной, логической и периферийной машин.

При выборе архитектуры центрального процессора рассматривался вариант ЦП АС-6, развитый в проекте БЭСМ-10, и векторно-конвейерный процессор. Достижение максимальной производительности было

возможно только при использовании синхронных конвейерных схем, более перспективной была признана векторно-конвейерная архитектура. Подсистема внешней полупроводниковой памяти отличалась наличием интеллектуального контроллера, предназначенного для реализации различных методов доступа к внешней памяти со стороны основной машины и обеспечивающего объединение двух векторных машин в одной системе. Пиковая производительность двухпроцессорной системы составила 500 Mflops [3]. Программное обеспечение состояло из операционных систем основной и внешних машин, систем программирования на языках макроассемблера, «Фортран 77», «Паскаль», «Си».

В 1991 г. были проведены испытания системы «Электроника СС БИС-1», изготовлены и налажены 4 образца, началась их установка у заказчиков. В том же году был разработан проект системы «Электроника СС БИС-2», направленный на создание многопроцессорной системы с производительностью до 10 Gflops. Кроме многопроцессорных основных машин планировалось включить в систему мониторные машины для управления системой и подготовки задач, а также подсистему с массовым параллелизмом. Однако в 1993 г. было принято решение о прекращении работ.

Опыт, накопленный при разработке указанных систем, позволил начать исследования по неоднородным вычислительным суперсистемам. Было показано, что более эффективным является такое построение системы, при котором в ней имеются тесно связанные подсистемы с различной архитектурой, ориентированные на разные формы параллелизма, имеющиеся в больших задачах. При обработке больших массивов данных имеет место параллелизм на уровне данных. С наибольшей эффективностью эта форма параллелизма используется в векторных машинах. Другая форма параллелизма – параллелизм на уровне задач – имеет место в программах, разбиваемых на большое число независимых или слабо связанных подзадач. В этом случае эффективнее многопроцессорная подсистема.

На первом этапе были разработаны концепции построения неоднородных суперсистем, в частности объединение в едином вычислительном модуле векторного унипроцессора с мультипроцессором на микропроцессорах на основе доступа к общей памяти [4]. В систему могут входить несколько таких модулей, объединенных в свою очередь общей системной памятью [5].

Следующий этап исследований был связан с анализом использования перспективных СБИС с очень высокой степенью интеграции. Использование принципа близкодействия обеспечивает достижение максимальной производительности при построении масштабируемого модульного мультиконвейерного процессора, в котором при выполнении сложных векторных операций возможно получение десятков или сотен результатов операций с плавающей запятой в один такт [6].

В соответствии с рассмотренными принципами был разработан исследовательский проект мультиархитектурной вычислительной суперсистемы, ориентированный на создание систем с производительностью более 1 Pflops [7]. Суперсистема состоит из вычислительной подсистемы, мониторно-моделирующей подсистемы, системной и внешней памяти. Вычислительная подсистема включает векторный мультипроцессор, скалярный мультипроцессор и мониторную машину. Разработан набор модулей для построения процессоров разных типов. Сравнение данного проекта с зарубежными исследованиями и разработками показывает, что имеет место опережение по концептуальным подходам к созданию суперсистем и использованию СБИС с предельными характеристиками [8].

### **Библиографический список**

1. Кузьмичев Д.А., Мельников В.А., Митропольский Ю.И., Смирнов В.И. Принципы составления документации для больших вычислительных машин // Доклад на Юбилейной научно-технической конферен-

- ции, посвященной 25-летию Института точной механики и вычислительной техники АН СССР, ИТМ и ВТ АН СССР. М., 1975. С. 3–16.
2. Мельников В.А., Митропольский Ю.И., Малинин А.И., Романков В.М. Требования к конструкции высокопроизводительных ЭВМ и проблемы ее реализации // Вопросы кибернетики: Комплексное проектирование элементарно-конструкторской базы суперЭВМ / Под ред. В.А. Мельникова и Ю.И. Митропольского. М., 1988. С. 3–10.
  3. Мельников В.А., Митропольский Ю.И., Шнитман В.З. Научные, технологические и методические аспекты создания вычислительной системы «Электроника СС БИС-1» // Юбилейный сборник трудов Отделения информатики, вычислительной техники и автоматизации Российской академии наук. М., 1993. С. 28–41.
  4. Митропольский Ю.И. Концепции построения неоднородных вычислительных суперсистем // Распределенная обработка информации: Труды Пятого Международного семинара / Институт физики полупроводников СО РАН, Новосибирск, 1995. С. 42–46.
  5. Анохин А.В., Ленгник Л.М., Митропольский Ю.И., Пучков И.И. Архитектура неоднородной вычислительной суперсистемы // Распределенная обработка информации. С. 22–27.
  6. Митропольский Ю.И. Архитектура мультимикровейерного модульного масштабируемого унипроцессора // Распределенная обработка информации: Труды Шестого Международного семинара / Институт физики полупроводников СО РАН. Новосибирск, 1998. С. 30–34.
  7. Митропольский Ю.И. Мультиархитектурная вычислительная суперсистема // Труды Первой Всероссийской научной конференции «Методы и средства обработки информации». М., 2003. С. 131–136.
  8. Митропольский Ю.И. Мультиархитектура – новая парадигма для суперкомпьютеров // Электроника: наука, технология, бизнес. 2005. № 3. С. 42–47.

## **СОЗДАНИЕ СИСТЕМЫ ПРОТИВОРАКЕТНОЙ ОБОРОНЫ (ПРО) И СУПЕРЭВМ» (ПО МАТЕРИАЛАМ ДОКЛАДА АКАДЕМИКА РАН В.С. БУРЦЕВА)**

**Ю.Н. Никольская**

Институт проблем информатики РАН

Москва, РФ

[unn@bur.oivta.ru](mailto:unn@bur.oivta.ru)

Доклад академика РАН В.С. Бурцева (1927–2005) «Создание системы противоракетной обороны (ПРО) и суперЭВМ», который был представлен на конференции «40-летие первого поражения баллистической ракеты средствами ПРО», посвящен созданию трех поколений управляющих вычислительных систем для противоракетной обороны СССР, работающих в режиме реального времени. В докладе подробно рассматриваются принципы организации управляющих вычислительных систем ПРО, созданных на основе разработанных ЭВМ (М-40, М-50, 5Э926, 5Э51, МВК «Эльбрус-1» и МВК «Эльбрус-2»), отмечаются особенности и архитектуры этих систем, обеспечение надежности их работы.

## РАЗВИТИЕ ПРОМЫШЛЕННОСТИ СРЕДСТВ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ В БЕЛОРУССКОЙ ССР

**В. В. Пржиялковский, Ю.В. Карпилович, Г.Д. Смирнов**

ОАО «НИЦЭВТ»

Москва, РФ

*przj@nicvt.ru*

В 60-х годах прошлого столетия в Белоруссии вырос комплекс предприятий, обеспечивший выпуск более 65% советских универсальных ЭВМ типа Mainframe.

Такой бурный рост производства ЭВМ и самих предприятий стал возможным в результате целого ряда мероприятий, целенаправленно проведенных руководством Минского завода им. Г.К. Орджоникидзе, техническими подразделениями завода и коллективом НИИЭВМ, выросшего из СКБ завода. Эти мероприятия были энергично поддержаны руководством Белорусской ССР и Министерством радиопромышленности СССР.

Важным аспектом при крупносерийном производстве ЭВМ является разработка моделей, пользующихся популярностью у пользователей и обладающих технологичной конструкцией, отвечающей требованиям крупносерийного производства.

В докладе излагаются основные организационно-технические мероприятия, проведенные Минским заводом им. Г.К. Орджоникидзе с целью наращивания производства ЭВМ. Рассматриваются архитектурные и конструктивно-технологические особенности ЭВМ «Минск-XX» и ЕС-10XX, обеспечившие возможность их крупносерийного производства Минским производственным объединением вычислительной техники.

Приводятся сравнительные данные по выпуску ЭВМ Минским объединением и другими заводами Министерства радиопромышленности.

## О ПИОНЕРСКИХ РАБОТАХ КИЕВСКОЙ ЛАБОРАТОРИИ С.А. ЛЕБЕДЕВА ИНСТИТУТА ЭЛЕКТРОТЕХНИКИ АН УССР И ИНСТИТУТА КИБЕРНЕТИКИ ИМ. В.М. ГЛУШКОВА АН УССР В ОБЛАСТИ СОЗДАНИЯ И РАЗВИТИЯ КОМПЬЮТЕРОВ

**З.Л. Рабинович, Б.Н. Малиновский**

Институт кибернетики им. В.М. Глушкова НАН Украины

Киев, Украина

*icfst@icfst.kiev.ua*

Эти разработки осуществлялись главным образом в следующих направлениях, определяемых типами компьютеров:

- 1) универсальные,
- 2) специализированные,
- 3) управляющие, широкого назначения,
- 4) интеллектуализированные ЭВМ с повышенным машинным (внутренним) интеллектом,
- 5) микро-ЭВМ, широкого назначения,
- 6) интеллектуализированные мультимикропроцессорные суперЭВМ, обладающие высокой и сверхвысокой производительностью за счет распределенной (мультипрограммной и распараллеленной) обработки информации в них.

Первое направление базировалось на фундаментальном изобретении С.А. Лебедевым электронной цифровой вычислительной машины, первый экземпляр которой (МЭСМ) был запущен в исследовательскую эксплуатацию в ноябре 1950 года и сдан уже в усовершенствованном виде Государственной комиссии 25 декабря 1951 года.

В докладе отмечаются наиболее характерные пионерские разработки по остальным направлениям, пользовавшимся известностью в СССР и за его пределами.

Наряду с констатацией исторически значимых достижений приводятся и некоторые соображения в отношении дальнейших перспектив в развитии указанных выше классов компьютеров.

Прикладные разработки дополнялись, как правило, созданием теории ЭВМ, которая, с одной стороны, обобщала опыт выполнения работ, а с другой – открывала дальнейшие пути и перспективы развития компьютерной науки и техники. Теоретические исследования получили отражение в ряде монографий, в том числе и фундаментальных, широко использовавшихся в НИИ и вузах.

За исключением первых работ по первому и второму направлениям, относящихся к научной школе С.А. Лебедева, остальные относятся к научной школе В.М. Глушкова, вернее, к собственно компьютерной ветви этой школы. Между участниками обеих школ всегда существовали дружеские деловые контакты и схожесть взглядов. Эти контакты не прерывались и после ухода из жизни этих великих ученых в апогее их творческой деятельности.

Работы в указанных направлениях интенсивно развиваются в Институте кибернетики имени В.М. Глушкова и сейчас, причем в значительной степени в духе идей и предначертаний, оставленных Глушковым. Так, например, в шестом направлении частично уже выполнены и продолжают развиваться работы по созданию ряда высоко- и сверхвысокопроизводительных компьютеров (суперЭВМ) с кластерной архитектурой и повышенным уровнем машинного интеллекта.

В последнее время проводятся также исследования бионического подхода к созданию «думающих» машин на основе вырабатываемых концептуальных представлений о структурах и процессах обработки информации в человеческом мышлении. Такого рода представления помимо прикладного значения в искусственном интеллекте имеют существенное познавательное значение в сфере исследований мозга как одной из главнейших научных проблем современности.

Дальнейшее изложение материала доклада осуществляется в пределах каждого указанного направления в приведенной исторической последовательности. Причем авторы, будучи участниками этих событий, основываются как на известных исторических и архивных материалах, так и на собственных воспоминаниях, в том числе и уже отраженных в литературе.

### **Универсальные ЭВМ**

Это направление работ Киевской лаборатории академика Сергея Алексеевича Лебедева относится к первой странице мировой истории зарождения и развития компьютеров.

Под руководством Сергея Алексеевича и с его личным основополагающим творческим и практическим на всех этапах работы участием была создана первая в СССР и континентальной Европе ЭВМ – так называемая Малая электронная счетная машина – МЭСМ. Разработка и проектирование собственно машины были начаты в лаборатории осенью 1948 года. Но самые общие принципы построения и работы ЭВМ, названных впоследствии компьютерами, были изобретены и сформулированы лично Сергеем Алексеевичем значительно раньше, чему имеются конкретные свидетельства. Поэтому эти принципы, как и архитектура первых однопроцессорных ЭВМ, называемые фон-неймановскими, вполне можно называть и лебедевскими.

ми. Причем это нисколько не умаляет роли и заслуг современников С.А. Лебедева, также одновременно с ним выдвигавших идеи и принципы построения ЭВМ и руководивших первыми разработками. (В хронологическом порядке ими в первую очередь являются И.С. Брук, Б.И. Рамеев, Ю.Я. Базилевский.)

Но, как бы то ни было, первой созданной в СССР ЭВМ оказалась МЭСМ. В своем начальном варианте она была сделана всего за два года причем буквально с нуля, поскольку никаких, даже самых элементарных базисных конструктивов не существовало. Шестого ноября 1950 года уже состоялся первый запуск машины на выполнение тестов, что было отмечено специальным актом в рабочей тетради С.А. Лебедева, подписанным им, партгором лаборатории Е.А. Шкабарой и профгором З.Л. Рабиновичем, удостоверяющими этим актом выполнение социалистического обязательства лаборатории к 26-й годовщине Великой Октябрьской социалистической революции (таковы были тогда обычаи, причем совсем не плохие). На данной стадии разработки МЭСМ она была выполнена лишь в виде действующего макета, на котором отрабатывались принципы ее эксплуатации и улучшались производственные характеристики, что позволило уже в 1951 году преобразовать макет в эффективно (по тем меркам) работающую ЭВМ, вполне пригодную для требуемой регулярной эксплуатации. 25 декабря 1951 года Государственная комиссия, возглавляемая академиком М.В. Келдышем, составила и подписала акт приемки машины. Эта дата и является официально зарегистрированной датой создания первой отечественной ЭВМ и началом ее регулярной эксплуатации. Одновременно с этим осуществлялось также дальнейшее повышение возможностей машины, в частности введение в нее дополнительной памяти на магнитном барабане, усовершенствование ввода-вывода и т.д. (что само по себе представляло уже определенные новшества). Особо важным фактором представляется то, что архитектура МЭСМ обладала всеми главными принципиальными чертами и качественными характеристиками, необходимыми для создания впоследствии высокопроизводительных ЭВМ в классе универсальных однопроцессорных машин. Ограниченность же ее количественных характеристик обуславливалась не только специфическими условиями того времени, но и необходимостью возможно быстрее создать первую в Советском Союзе ЭВМ.

Таким образом, разработка МЭСМ явилась первым основополагающим этапом деятельности С.А. Лебедева по осуществлению главной цели его творчества, а именно создание отечественной высокопроизводительной электронной вычислительной техники. Первой ЭВМ этого класса стала машина БЭСМ, положившая начало знаменитой серии этих машин. Их архитектурные принципы, ставшие классическими, в самом начальном виде были опробованы и исследованы еще в опытной эксплуатации машины МЭСМ, использованной таким образом в качестве действующего официального макета БЭСМ.

Весьма знаменательно, что принципы построения машины МЭСМ оказались более прогрессивными по сравнению с аналогичными, осуществленными в первых английских и американских ЭВМ. В частности, в МЭСМ была впервые реализована параллельная обработка кодов в процессоре, а во внутреннем языке машины и системе ее управления были предусмотрены средства преобразования компонентов программ. МЭСМ, фактически, стала одной из самых первых в мире ЭВМ, отвечающей названию «компьютер».

Сразу же после приема МЭСМ Государственной комиссией, т.е. с 1952 года, она начала использоваться в регулярной эксплуатации для решения сложных народнохозяйственных и оборонных задач, будучи в то время единственной пригодной для этого в Союзе. Среди этих задач особо можно отметить баллистические расчеты, выполненные в 1952–1953 годах группой математиков под руководством А.А. Ляпунова из Математического института Академии наук СССР.

Наряду с пользовательской эксплуатацией МЭСМ широко использовалась для разработки методов и языков программирования и первых информационных технологий, а также для определения путей разви-

тия самой компьютерной техники, в том числе и для исследования возможности перевода новых ЭВМ на безламповую элементно-конструктивную базу (Б.Н. Малиновский).

Опыт создания и работы МЭСМ существенно способствовал развитию цифровой вычислительной техники в СССР, большим научным вкладом в этом смысле явилась монография С.А. Лебедева, Л.Н. Дашевского и Е.А. Шкабары «Малая электронная счетная машина» (М., 1952). Книга была издана под грифом «секретно» и предназначалась для специалистов ВТ.

Следующей пионерской и исключительно важной работой лаборатории в области универсальных ЭВМ было создание ЭВМ «Киев». Эта работа была начата уже после окончательного переезда С.А. Лебедева в Москву. Лабораторию С.А. Лебедева в это время возглавил академик Борис Владимирович Гнеденко. Разработка ЭВМ «Киев» была завершена при появлении Виктора Михайловича Глушкова, ставшего в 1956 году заведующим лабораторией. Главным конструктором разработки был Лев Наумович Дашевский. ЭВМ «Киев» следует отнести к пионерским работам, поскольку в ней были применены новые оригинальные архитектурные решения – асинхронное управление основными устройствами, групповые операции с модификацией адресов, выполняемых над сложными структурами данных и др., а также ряд элементно-конструктивных усовершенствований, включая ферритовую оперативную память, внешнее ЗУ на магнитном барабане, ввод-вывод чисел в десятичной системе счисления и др. (некоторые из них были применены в ранее созданной специализированной ЭВМ СЭСМ – см. далее).

ЭВМ «Киев» стала основным оборудованием Вычислительного центра Академии наук УССР, образованном на основе лаборатории, оборудование которой до этого составляли две машины – МЭСМ и СЭСМ. Также как и при использовании этих машин, на машине «Киев» помимо реализации вычислений весьма интенсивно проводились научные исследования с целью выработки первых информационных технологий, методов взаимодействия пользователей с машиной, усовершенствование алгоритмических языков программирования и др. Причем эти работы велись уже в процессе создания самой машины и оказали определенное влияние на ее архитектуру.

В монографиях Дашевского Л.Н., Погребинского С.Б., Шкабары Е.А. «Вычислительная машина "Киев"» (Киев, 1964); Глушкова В.М., Ющенко Е.Л. «Вычислительная машина "Киев"» (Киев, 1962) указанные исследования обобщены и стали существенным научным вкладом в развитие компьютерной науки.

ЭВМ «Киев» стала известной не только в Украине. Ее специально изготовленный второй экземпляр был использован в качестве основной машины Вычислительного центра Международного института ядерных исследований в г. Дубне.

### **Специализированная ЭВМ**

Сергей Алексеевич Лебедев первым выдвинул идею создания специализированной электронной счетной машины (СЭСМ) для решения систем линейных алгебраических уравнений итерационными методами, в которой расчеты полностью были бы совмещены с вводом исходных данных. При этом оказалось возможным применение последовательного арифметического устройства с использованием в нем специальных динамических регистров на магнитном барабане, на котором было осуществлено и основное ЗУ машины. Столь рациональный главный принцип структуры машины, предложенный Сергеем Алексеевичем (одновременность ввода и расчетов и последовательная арифметика), обеспечил относительно весьма малые аппаратные затраты в машине, чему еще способствовала реализация логических элементов преимущественно на ферритных трансформаторах и полупроводниковых диодах. Это обстоятельство (примерно в 10 раз меньше ламп, чем в МЭСМ) особо акцентировалось в опубликованной в американском журнале «Data-

mation» статье с весьма похвальным отзывом о СЭСМ. Более того, в США даже была переиздана на английском языке коллективная монография «Специализированная электронная счетная машина СЭСМ», написанная разработчиками машины З.Л. Рабиновичем (главный конструктор СЭСМ), Ю.В. Благовещенским, Р.Я. Черняком и др. (издана издательством Академии наук УССР под редакцией В.М. Глушкова в 1961 году). Эта книга явилась одной из первых книг по цифровой вычислительной технике, переведенных и изданных за рубежом, и поддержала многие актуальные в то время теоретико-технические положения: мелкоблочная конструкция СЭСМ, передовая для того времени; отсутствие обычного программирования, поскольку нужные режимы вычислений устанавливались пакетными переключателями и осуществлялись встроенным управлением, исходные же данные и результаты выражались в десятичной системе; режимы машины по сути представляли собой различные групповые операции, которыми осуществлялись не только собственно итерация при решении систем линейных алгебраических уравнений, но и, например, подсчет корреляционных функций и выполнение различных матрично-векторных операций.

Таким образом, СЭСМ уже содержала в своей структуре прототип матрично-векторного процессора, вошедшего в практику построения ЭВМ несколько позднее.

СЭСМ регулярно эксплуатировалась в Вычислительном центре АН УССР, а затем была передана в учебное заведение для лабораторных работ учащихся.

Несмотря на эффективность операционной специализации, доминирующее развитие получили все же универсальные ЭВМ. При этом преимущества такой специализации не были утеряны, поскольку они воплощались в функциональных структурных узлах этих ЭВМ (например, матрично-векторных процессорах). Но все же специализация ЭВМ, уже на базе универсальных машин, была не только сохранена, но и существенно развита, а именно в направлении их ориентации на функциональные предназначения.

### **Первая полупроводниковая цифровая управляющая машина широкого назначения**

Научно-технический прогресс конца 50-х годов прошлого века поставил задачу создания ЭВМ для управления (непосредственного или опосредованного) промышленными объектами, анализа процессов, происходящих в наблюдаемой среде и т.п. Фундаментальным принципом построения таких ЭВМ должно было стать их системное оборудование, содержащее как внутреннее математическое обеспечение, так и (что особенно характерно) структурные устройства для получения данных от управляемых либо подвергаемых анализу объектов и обратного прямого воздействия на них или при моделировании их с целью изучения.

Идея создания управляющей машины широкого назначения была выдвинута Виктором Михайловичем Глушковым в 1957 году. Создание такой машины было истинно пионерской разработкой, научное руководство которой Виктор Михайлович разделил с Б.Н. Малиновским, ставшим также главным конструктором машины.

Широкое назначение УМШН было обеспечено предварительным и сопутствующим ее проектированию и использованию исследованием специфических особенностей существенно различных управляемых производственных объектов и технологических процессов. Исследование машины в лабораторных и производственных условиях показало ее высокие эксплуатационные качества и надлежащую эффективность использования, что подтвердило рациональность принципа широкого назначения для управляющих машин и позволило рекомендовать УМШН, получившей название «Днепр», для серийного производства.

Принцип широкого назначения обусловил следующие основные характеристики управляющей машины:

- алгоритмически полный и, более того, существенно развитый набор операций во внутреннем языке;
- гибкое программное управление;

- секционное построение запоминающего устройства для возможности выбора пользователем желаемого объема памяти;
- возможность подключения к машине широкого назначения различных источников входной информации (в том числе и аналоговых);
- возможность подключения к машине широкого назначения печатающих устройств для вывода получаемых результатов.

Все эти структурные характеристики были обеспечены в УМШН «Днепр». Она обладала также относительно высокими для того времени конструктивными и количественными характеристиками (10 тыс. оп./сек) и стала первой в СССР машиной, получившей массовое использование при автоматизации технологических процессов, сложных физических экспериментов, объектов военного назначения и др. В Институте кибернетики АН УССР был разработан первый в стране гибридный комплекс, сочетающий управляющую машину УМШН «Днепр» с аналоговой машиной ЭМУ-10.

Значение создания УМШН «Днепр» этим не ограничилось, поскольку разработка машины существенно повлияла на развитие отечественного вычислительного машиностроения вообще и на развитие кибернетической техники в частности. Она выпускалась в течение 10 лет, чем был установлен рекорд долголетия среди подобных машин. Опыт создания и использования УМШН «Днепр» отражен в монографии Б.Н. Малиновского «Цифровые управляющие машины и автоматизация производства» (М., 1963) широко использовавшейся в проектно-конструкторских и учебных организациях, а также в книге «Нет ничего дороже...» (Киев, 2005).

### **Интеллектуализированные ЭВМ**

Необходимость и развитие массового использования цифровой вычислительной техники привело к созданию мини-ЭВМ, предназначенных для личного использования в режиме непосредственного и достаточно простого для массового пользователя взаимодействия с машиной, в том числе и прямого диалога с ней.

Достижение достаточной универсальности таких машин, учитывая существенную ограниченность их чисто аппаратных ресурсов, было обеспечено гибкими микропрограммными средствами, которыми их структуры динамически адаптировались под решаемую задачу.

По инициативе В.М. Глушкова в Институте кибернетики и его СКБ была разработана первая в Советском Союзе мини-ЭВМ «Промінь» с многоступенчатым микропрограммным управлением. Эта машина, будучи запущенной в серийное производство, была, как упоминал Виктор Михайлович, «по сути новым словом в мировой практике». Ее характерным достоинством была реализация ряда часто встречающихся процедур, таких, например, как выполнение макрокоманд, микропрограммы которых хранились в ЗУ. По сути это означало повышение уровня внутреннего, т.е. интерпретируемого в процесс вычислений языка машины, что избавляло пользователей от необходимости программирования этих процедур. Во второй половине 70-х годов прошлого века мини-ЭВМ «Промінь» и «МИР» (разработанный через три года после ЭВМ «Промінь» – см. далее) составили почти треть выпускаемых в СССР ЭВМ.

После пионерских разработок УМШН «Днепр» и ЭВМ «Промінь» в Институте кибернетики АН УССР возобладала генеральная линия создания ЭВМ с повышенным машинным интеллектом. Некоторые предпосылки которого уже были отражены в этих машинах. Само же это понятие появилось несколько позже, и его введение устранило расплывчатость термина «интеллектуализация ЭВМ». Работы развернулись с выдвинутого «дерзкой» (по выражению В.М. Глушкова) идеи как можно большего приближения внутреннего (машинного) языка к математическому, т.е. придания ему принципиальных свойств алгоритмических язы-

ков программирования, еще достаточно высокого уровня. Такой внутренний язык предполагал структурно поддерживаемую эффективную систему интерпретации.

Эта идея была реализована в пионерских разработках Института кибернетики АН УССР – серии мини-ЭВМ «МИР» и проекте высокопроизводительной ЭВМ «Украина». Эти работы вызвали весьма большой резонанс и признание в СССР и за рубежом. ЭВМ «МИР» была единственной советской машиной, закупленной в США (с целью подтверждения советского приоритета, в чем фирма IBM была заинтересована по деловым обстоятельствам). Весьма характерным является тот факт, что указанным разработкам предшествовала глубокая теоретическая подготовка, детализирующая основные идеи. Необычность идей, заложенных в МИРы, как раз и вызывала далеко не единодушное их признание, подчас даже и ожесточенные споры.

Но как только была создана машина «МИР-1» (научный руководитель В.М. Глушков, главный конструктор С.Б. Погребинский) и обнародован опыт ее эксплуатации в отношении целесообразности мини-ЭВМ с микропрограммной реализацией в ней внутреннего языка высокого уровня, сомнения исчезли, а в отношении применения его в высокопроизводительных (т.е. мощных) ЭВМ они оставались. Однако эти сомнения в результате публичной защиты в МРП проекта ЭВМ «Украина», а также возникновения подобных разработок в некоторых ведущих организациях России и Армении сменились уверенностью в правильности и эффективности такого рода интеллектуализации ЭВМ. Весьма высокий уровень ЭВМ семейства «МИР» обеспечивался в первую очередь следующими архитектурными характеристиками и возможностями машины:

- весьма высокий уровень внутреннего языка, в котором машинные операции прямо соответствовали операторам пользовательского языка, а этот входной язык – «аналитик» – был специально ориентирован на инженерные расчеты, включая и сложные аналитические преобразования;
- развитые системы структурной интерпретации внутреннего языка с использованием динамических магазинных автоматов в памяти;
- произвольные точность и диапазоны вычислений путем динамического регулирования количества разрядов чисел, а также их диапазонов.

Приведенные и другие характерные архитектурные и структурные особенности машины семейства «МИР», основанные на иерархическом микропрограммном управлении в них, вполне позволяют трактовать значение этой машины (как и ЭВМ «Промінь») как прототипа впоследствии появившихся персональных ЭВМ. Более того, МИРы по некоторым характеристикам уровня интеллектуальности существенно превосходили появившиеся ПЭВМ.

Признание и весьма высокая оценка этой знаменательной пионерской разработки ознаменовалась присуждением ее авторам Государственной премии СССР. Однако отсутствие во время создания ЭВМ «Промінь» и «МИР» надлежащих базисных конструктивных и технологических возможностей в СССР не позволило превратить МИРы в конструктивно совершенные персональные ЭВМ. В американской научной печати даже указывалось, что современные конструктивы еще пока не позволяют решить важную проблему повышения уровня внутренних языков до алгоритмических входных. И если для машин «МИР» удалось это сделать, то для высокопроизводительной ЭВМ с параллельной обработкой данных это оказалось в то время практически неосуществимым. Именно поэтому был разработан в Институте кибернетики АН УССР проект высокопроизводительной ЭВМ «Украина», который хоть и не был реализован в «металле», но послужил для доказательства возможностей и целесообразности реализации в таких ЭВМ высокоуровневых внутренних языков.

Основные принципы построения ЭВМ «Украина» заключались в следующем:

- семантическая тождественность ее внутреннего языка алгоритмическому языку АЛГОЛ-60 (как высокоуровневому в то время языку для универсальных вычислений, причем еще обогащенному логическими операциями);
- применение виртуальной памяти динамически распределяемой и автоматически проецируемой на физическую память;
- методов структурной интерпретации, эффективно поддерживаемой аппаратными средствами (в том числе магазинными автоматами);
- широким применением методов ускоренного (в том числе распараллеленного) выполнения различных операций;
- предусмотрением в штатном математическом обеспечении широкого набора непрограммируемых пользователем сложных процедур, а также подпрограмм реализации типовых компонент различных вычислительных процессов.

Также как и создание ЭВМ «МИР», материалы по проекту ЭВМ «Украина» (главный конструктор З.Л. Рабинович), разосланные в виде опубликованного научно-технического отчета в авторитетные организации плодотворно повлияли на развитие высокопроизводительной вычислительной техники в СССР. Так, в книге по ЭВМ «Эльбрус» прямо сказано, что этот суперкомпьютер был построен на двух фундаментальных принципах – архитектуре БЭСМ-6 и принципе структурной интерпретации языков высокого уровня (со ссылкой при этом на В.М. Глушкова, машины «МИР» и проект машины «Украина»).

Мировой престиж разработки «Украина» защищен авторским свидетельством с приоритетом от 9 мая 1962 года (но выданным лишь в 1968 году после его детального обсуждения и защиты в соответствии с весьма положительным экспертным заключением академика АН СССР А.П. Ершова). Хочется особо отметить, что идеология развития вычислительной техники СССР не отставала от передовых западных стран (включая США), но технологические возможности у нас далеко не всегда были в наличии.

В распространении и последующем развитии идей интеллектуального развития ЭВМ немалое значение имели публикации ряда статей и монографий, из которых укажем две: «Вычислительные машины с развитыми системами интерпретации» В.М. Глушкова, А.А. Барабанова, Л.А. Калиниченко, С.Д. Михновского, З.Л. Рабиновича (Киев, 1970) и «Машинный интеллект и новые информационные технологии» Т.А. Грипченко, А.А. Стогния (Киев, 1993). Первая ориентирована была на опыт проекта «Украина» и ввиду ее общетеоретического значения широко использовалась в вузах. Во второй монографии был обобщен опыт разработки МИРов и развиты актуальные и сейчас методы эффективного использования ЭВМ, которому весьма благоприятствует высокий уровень машинного интеллекта.

В заключение заметим, что с повышением производительности ЭВМ с параллельной архитектурой (на основе распределенной обработки информации) машинный интеллект приобретает новые специфические черты, о чем концептуально сказано в статье З.Л. Рабиновича «О концепции машинного интеллекта и его развитии» (Кибернетика и системный анализ. 1995. № 2. С. 163–179).

### **Первое отечественное семейство микроЭВМ широкого назначения**

К этому времени уже появилась возможность на основании достижений электронной технологии перейти и к созданию микроЭВМ. Впервые работы были развернуты в Ленинградском СКТБ при НПО «Светлана». Институт кибернетики АН УССР принял активное участие в их создании, а именно в части идеологии построения ее архитектуры (Б.Н. Малиновский, А.В. Палагин). В 1976 году в ЛКТБ «Светлана» было создано

первое отечественное семейство из трех микроЭВМ широкого назначения «Электроника С5» (С5-01, С5-11, С5-21) для измерения, контроля и управления процессами и объектами. МикроЭВМ обладали микропрограммным управлением для выполнения сложных групповых операций (процедур), представляли собой микропроцессорные комплексы с возможностями применения их в различных конфигурациях. В микроЭВМ был использован микропроцессор с гибкой архитектурой. Он представлял собой однокристалльный 16-разрядный эмулирующий микропроцессор. В США этот результат был повторен позднее, а наше достижение (автор А.В. Палагин) в одном из американских журналов было названо «русским вызовом».

Немалое значение в развитии теории ЭВМ имело обобщение итогов этой разработки (а также других) в монографии Палагина А.В., Денисенко Е.Л., Белицкого Р.И. и др. «Микропроцессорные системы обработки информации» (Киев, 1993).

### **Высоко- и сверхвысокопроизводительные ЭВМ с повышенным машинным интеллектом**

Для построения и эксплуатации высоко- и сверхвысокопроизводительных ЭВМ весьма эффективным оказался выдвинутый В.М. Глушковым принцип макроконвейерной обработки данных, которая заключена, по его же определению, «в том, что каждому отдельному процессору на очередном шаге вычислений дается такое задание, которое позволяет ему длительное время работать автономно, без взаимодействия с другими процессорами». Этот принцип был полностью реализован в пионерской разработке многопроцессорной системы «Макроконвейер», которая в отличие от имевшихся в то время многопроцессорных систем обладала весьма эффективным свойством – почти линейной зависимостью производительности от количества процессоров в существенно большем диапазоне. Система «Макроконвейер» была создана в двух серийных вариантах ЕС-2701 и ЕС-1766 (производительность второго при использовании полного комплекта процессоров – 256 – оценивалась в 128 миллионов оп./сек). В то время (80-е годы прошлого века) это были, по видимому, самые мощные в СССР многопроцессорные ЭВМ.

Виктор Михайлович уже не смог увидеть эти макроконвейерные ЭВМ (главный конструктор С.Б. Погребинский), но научный вклад в их создание был им в литературе сделан, а затем развит и продолжен его последователями.

Весьма характерным является то обстоятельство, что в целях всемерного повышения производительности вычислительных систем макроконвейерные системы с относительно простыми процессорами, но с большим их количеством оказываются гораздо более перспективными, чем комплекс на мощных процессорах, но при этом уже с их весьма ограниченным количеством. Это признали сами создатели таких комплексов (также подчас оказывающимися нужными в практике), по отношению к которым парадигма макроконвейерных ЭВМ является по сути альтернативной. И в дальнейших пионерских разработках института она получила весьма существенное развитие на базе уже использования совершенной микропроцессорной техники, позволившего создавать ЭВМ в виде мультимикропроцессорных с весьма большими количествами микропроцессоров в них. Но это в свою очередь потребовало применений так называемых кластерных архитектур, в которых задания в вычислительном процессе возлагались бы на отдельные кластеры, состоящие уже из нескольких микропроцессоров.

Первой пионерской разработкой в этом плане являлось создание мультимикропроцессорной ЭВМ нового класса – интеллектуальной решающей машины ИРМ, которая в себе сочетала распределенную, автоматически управляемую обработку информации с реализацией внутреннего языка высокого уровня, подобного С++ с введенными в него еще специальными графовыми операциями. Собственно такое сочетание

и определяло принадлежность этой ЭВМ к новому классу. Главной задачей на период конца прошлого и начала нынешнего века являлось всемерное повышение производительности ЭВМ (которое должно было выразиться в создании суперЭВМ мирового уровня), исходя из этого в Институте кибернетики имени В.М. Глушкова НАН Украины была совершена пионерская разработка мультимикропроцессорной суперЭВМ с кластерной архитектурой, макетом для создания которой и послужил действующий экземпляр упомянутой ЭВМ ИРМ (научный руководитель В.Н. Коваль, главный разработчик О.Н. Булавенко).

Уже в этом веке в Институте кибернетики имени В.М. Глушкова построены совместно с компанией ЮСТАР и эффективно эксплуатируются две модели указанной суперЭВМ – СКІТ 1 и СКІТ 2 с пиковой производительностью соответственно 170 и 350 гигафлопс. В скором времени предполагается создание модели с производительностью до 4 терафлопс. Так был совершен прорыв в области вычислительной техники в Украине, позволивший получить результат мирового уровня. Более того, в части машинного интеллекта предполагается его дальнейшее повышение (научный руководитель работы І.В. Сергієнко, главный конструктор В.Н. Коваль).

15 марта 2006 года Валерий Николаевич Коваль скоростно ушел из жизни. Триумфальный успех в достижении поставленной перед ним цели потребовал колоссального напряжения сил. Он буквально сгорел на работе. І.О. главного конструктора мультимикропроцессорных кластерных ЭВМ сейчас является А.А. Якуба.

Актуальность и дальнейшие планы развития этих работ отмечены в имеющихся публикациях:

1. Рабинович З.Л. О концепции машинного интеллекта и его развитии // Кибернетика и системный анализ. 1995. № 2. С. 163–179.
2. Коваль В.Н., Булавенко О.Н., Рабинович З.Л. Интеллектуальные решающие машины как базовые средства высокопроизводительных вычислительных систем // УСМ. 1998. № 6. С. 42–49.
3. Koval V., Bulavenko O., Rabinovich Z. Parallel Architectures and Their Development on the Basis of Intelligent Solving Machines // Proc. Int. Conf. on Parallel Computing in Electrical Engineering, Warsaw (Poland), 2002. P. 21–26.
4. Коваль В.Н., Савьяк В.В. Мультипроцессорные кластерные системы: планирование и реализация // Искусственный интеллект. 2004. № 3. С. 117–126.
5. SCIT – Ukrainian supercomputer project / Koval V.N., Ryabchun S.G., Savyak V.V., Sergienko I.V., Yakuba A.A. // Proc. International Conference KDS, 2005. Bulgaria, Varna, 2005.
6. Сергієнко І.В., Коваль В.Н. СКІТ – Український суперкомп'ютерний проект // Вісник Національної Академії наук України. 2005. № 8. С. 3–13.

Правительственная поддержка украинскому суперкомпьютерному проекту дает основание надеяться, что он будет полностью выполнен и это станет достойным научным вкладом для сохранения памяти о В.Н. Ковале.

## ЧЕТЫРЕ ПОКОЛЕНИЯ ВЫЧИСЛИТЕЛЬНЫХ МАШИН М.А. КАРЦЕВА (М-4, М4-2М, М-10, М-13)

**Ю.В. Рогачев**

НИИ вычислительных комплексов им. М.А. Карцева  
Москва, РФ

Михаил Александрович Карцев, крупный ученый в области вычислительной техники, доктор технических наук, профессор, главный конструктор четырех поколений ЭВМ и вычислительных комплексов для работы в режиме реального времени, основатель и первый директор НИИ вычислительных комплексов пришел в вычислительную технику еще будучи студентом радиотехнического факультета МЭИ. Первые понятия о принципах построения цифровых электронных вычислительных машин он получил от члена-корреспондента АН СССР И.С. Брука, в лаборатории которого в 1950 году он участвовал в создании ЭВМ М-1, проектируя для нее главный программный датчик (устройство управления). При создании следующей машины – ЭВМ М-2 в 1952–1953 годах он уже возглавлял группу инженеров и был признанным лидером разработки.

### **Первое поколение – ЭВМ М-4 и М4-М**

В 1957 году в Институте электронных управляющих машин (этот институт был образован на базе лаборатории И.С. Брука) разрабатывалось техническое задание на проектирование ЭВМ для экспериментального комплекса РЛС контроля космического пространства. ТЗ утвердили академик АН СССР А.Л. Минц и член-корреспондент АН СССР И.С. Брук. Руководство работами по созданию этой ЭВМ и вводу ее в эксплуатацию совместно с РЛС возложили на М.А. Карцева, образовав под его руководством специальную лабораторию № 2. Новой ЭВМ присвоили наименование М-4. Спецлаборатория № 2 явилась основой для развития коллектива и его роста до уровня Научно-исследовательского института вычислительных комплексов, а машина М-4 определила дальнейшее направление новых разработок М.А. Карцева – создание серии вычислительных машин типа «М» для построения вычислительных комплексов, обеспечивающих решение задач и обработку информации в реальном масштабе времени. М-4 открыла первое поколение машин Карцева такого класса. Авторский коллектив разработки: главный конструктор – к.т.н. М.А. Карцев, основные разработчики – Г.И. Танетов, Л.В. Иванов, Р.П. Шидловский, Ю.В. Рогачев, Г.И. Смирнова, Р.П. Макарова, Е.С. Шерихов, В.П. Кузнецова. Разработка и передача конструкторской документации заводу-изготовителю была завершена в 1958 году.

М-4 имела вполне современную на тот момент архитектуру ЭВМ для обработки информации в реальном масштабе времени и включала: центральную процессорную часть (устройство управления, арифметическое устройство, внутренние запоминающие устройства – оперативную память, постоянную память программ, констант, таблиц) и систему ввода-вывода информации.

Арифметическое устройство полупроводниковое: триггер строился на транзисторах П16, логические схемы на диодах Д9. Использовалась импульсно-потенциальная система элементов. Потенциальные элементы – диодная логика; импульсно-потенциальные элементы – клапан (двухвходовое И), один вход которого был импульсным, а второй вход – потенциальным. Триггер имел импульсные входы и потенциальные выходы. Основой АУ были однотипные ячейки, каждая из которых содержала один разряд обрабатываемой аппаратуры. В узле управления АУ находилось несколько триггеров, которые с помощью диодной логики при выполнении тех или иных арифметических операций сцеплялись в соответст-

вующие цепочки и кольца. В этом АУ была осуществлена впервые в мире аппаратная реализация операции вычисления квадратного корня.

Центральное управление предназначено для автоматического управления выполнением операций. Оно выбирает из постоянной памяти программ (ПП) инструкции и в соответствии с имеющимися в них указаниями выбирает необходимые константы из постоянной памяти констант (ПК) и данные, подлежащие обработке, из оперативной памяти; производит над этими данными те или иные действия и записывает результаты этих действий в указанные ячейки ОП. По указаниям программы эти результаты могут направляться на кодовые магнитные самописцы типа ПР-16, на устройство отображения информации, на печатающее устройство. Новые данные для обработки поступали в оперативную память машины непосредственно из устройства сопряжения с РЛС (при работе в реальном времени) или вводились с перфоленты при помощи фотоэлектрического считывающего устройства.

Оперативная память представляла собой ферритовое устройство матричного типа на 1024 24-разрядных чисел. В качестве запоминающего элемента использовались сердечники диаметром 1,5 мм (внутренний – 1,1 мм) и высотой 0,7 мм, изготовленные из магнитного материала ВТ-1. Цикл обращения – 1,4 мкс.

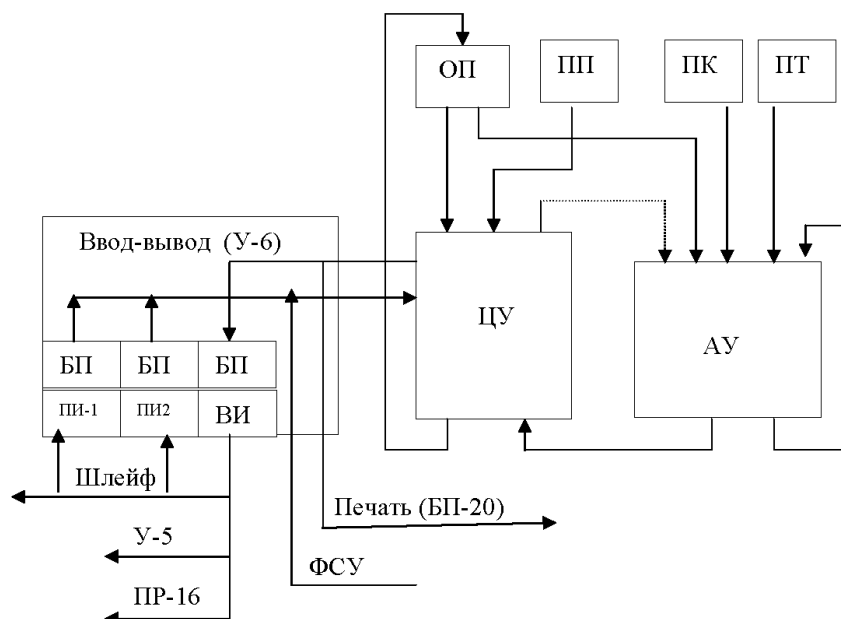
Постоянная память матричного типа, построена на тех же сердечниках, что и ОП. Различие только в порядке прошивки информационных обмоток.

Комплект внешних устройств включал фотосчитывающее устройство с перфолент (ФСЧУ), устройство «быстрой» печати (БП-20) и специализированное абонентское сопряжение для ввода и вывода информации РЛС с параллельным 14-канальным шлейфом. Для контроля информации использовались электронный индикатор для визуального отображения информации (У-5) и кодовые самописцы ПР-16 на магнитной ленте.

ЭВМ М-4 была асинхронной, имела одноадресную систему команд, большой набор арифметических операций, в том числе аппаратно реализованных сложных операций (умножение, двойное сравнение, извлечение квадратного корня и др.) и операций управления. Соблюдался принцип максимальной загрузки всего оборудования. Для преобразования адресов использовалось специальное 10-разрядное устройство. Резервы по производительности использовались для автоматического контроля работы машины путем двойных просчетов отдельных частей программы и запуска тест-программ.

### **Основные технические характеристики**

|                                              |                                                   |
|----------------------------------------------|---------------------------------------------------|
| Система счисления .....                      | двоичная                                          |
| Количество двоичных разрядов .....           | 23                                                |
| Представление чисел .....                    | с фиксированной запятой                           |
| Среднее быстродействие .....                 | 20 тыс. операций в секунду                        |
| .....                                        | 50 тыс. операций сложения или вычитания в секунду |
| .....                                        | 15 тыс. операций умножения в секунду              |
| .....                                        | 5,2 тыс. операций деления или                     |
| .....                                        | извлечения квадратного корня в секунду            |
| Объем оперативной памяти .....               | 1024 23-разрядных числа                           |
| Объем постоянной памяти программ .....       | 1024 23-разрядных числа                           |
| Объем постоянной памяти констант .....       | 128 12-разрядных чисел                            |
| Объем постоянной памяти таблиц .....         | 256 44-разрядных чисел                            |
| Объем буферной памяти узла ПИ-1 .....        | 120 15-разрядных чисел                            |
| Объем буферной памяти узла ПИ-2 .....        | 64 15-разрядных числа                             |
| Объем буферной памяти узла ВИ .....          | 1280 32-разрядных чисел                           |
| Ввод/вывод информации                        |                                                   |
| При работе в реальном масштабе времени ..... | параллельный, 14 каналов,                         |
| .....                                        | скорость 6250 чисел в секунду                     |
| Скорость ввода информации с перфоленты ..... | 45–50 чисел в секунду                             |
| Скорость печати .....                        | 7 строк по 6 шестнадцатеричных цифр в секунду     |



**Блок-схема ЭВМ М-4:**

*ЦУ* – центральное управление; *АУ* – арифметическое устройство; *ОП* – оперативная память; *ПП* – постоянная память программ; *ПК* – постоянная память констант; *ПТ* – постоянная память таблиц; *БП* – буферная память регистров; *ПИ-1*, *ПИ-2* – регистры приема информации; *ВИ* – регистр выдачи информации; *У-5* – устройство отображения информации; *ПР-16* – магнитные самописцы (магнитная лента); *БП-20* – печатающее устройство («быстрая печать»); *ФСУ* – фотосчитывающее устройство



**Общий вид ЭВМ М-4**

В 1960 году было изготовлено два комплекта этой машины. Первый комплект был направлен на объект для стыковки с РЛС экспериментального комплекса. Второй комплект предназначался для управления и обработки информации РЛС другого частотного диапазона, поэтому для него предстояло разработать и изготовить дополнительное устройство первичной обработки с более высоким быстродействием. Была разработана принципиально новая система потенциальных логических элементов с использованием высокочастотных диффузионных транзисторов. Основу системы составлял инвертор – формирователь уровней с многоступенными диодными логическими схемами. Функциональная схема устройства предусматривала построение широкоформатных регистров, которые обеспечивали одновременно параллельную обработку до 16 радиолокационных сигналов. Эти два новых технических решения обеспечили выполнение заданных требований по быстродействию. Устройство включало переключатель секторов, преобразователь кодов, накопитель, пороговое устройство, устройство перекодирования, буферную память. Все оборудование размещалось в одном типовом шкафу машины М-4. Устройство первичной обработки в 1961 году было изготовлено и состыковано со вторым комплектом М-4, который в этой комплектации получил обозначение М4-М и включался для совместной работы со второй РЛС на объекте.

В июле 1962 года первый комплект ЭВМ М-4 в составе экспериментального радиолокационного комплекса успешно выдержал совместные испытания. Оба комплекта этих машин работали совместно с радиолокационными станциями экспериментального комплекса до 1966 года.

### **Второе поколение – ЭВМ М4-2М и М4-3М**

В 1961–1962 годах в коллективе Карцева наряду с работами по машине М-4 продолжались исследования в поисках путей создания вычислительной машины с более высокими техническими характеристиками по производительности и надежности, более технологичными в изготовлении и удобными в эксплуатации. В начале 1963 года, когда Карцев получил задание на новую разработку, элементная база, конструкция, многие структурные и схемные вопросы были уже проработаны. Завод в этом же году получил полный комплект документации и смог приступить к подготовке производства. В соответствии с техническим заданием машина предназначалась для построения вычислительных комплексов территориальной вычислительной системы, включающей в свой состав несколько региональных вычислительных центров, связанных каналами передачи данных с главным вычислительным центром. Всем вычислительным комплексам ставилась задача обеспечивать постоянную круглосуточную непрерывную работу в режиме реального времени. Новая разработка получила обозначение М4-2М. Авторский коллектив разработки: главный конструктор – М.А. Карцев, заместители главного конструктора – Л.В. Иванов, Ю.В. Рогачев, Р.П. Шидловский, основные разработчики – Г.И. Танетов, В.А. Брик, Л.З. Либуркин, В.П. Кузнецова, Е.С. Шерихов, Е.И. Цибуль.

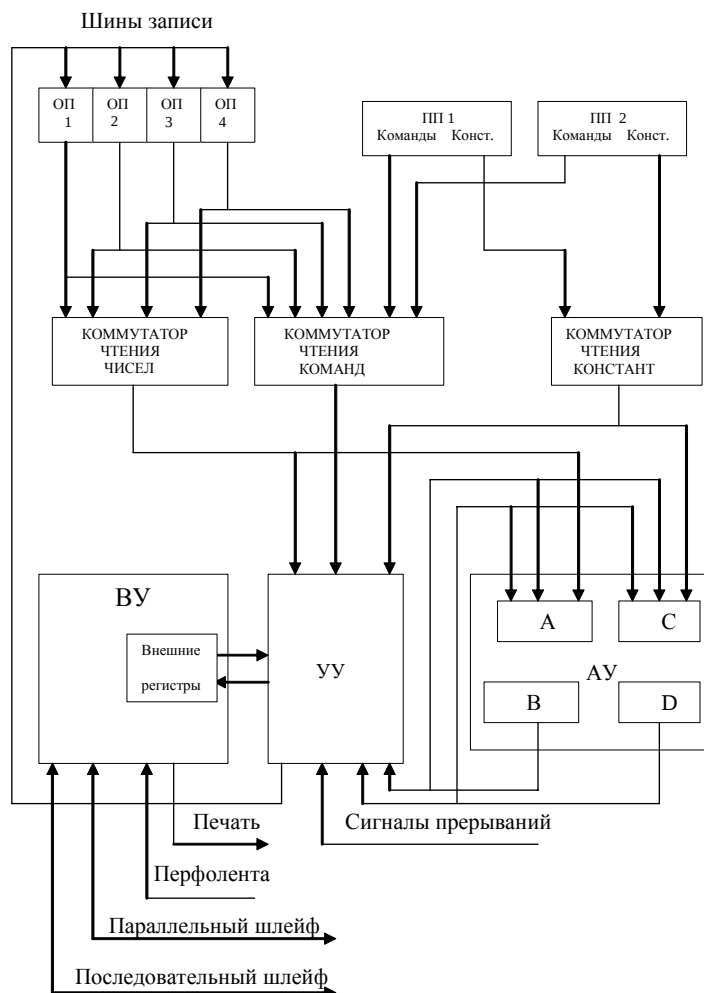
**ЭВМ М4-2М** имела одноадресную систему команд с возможностью чтения константы в качестве второго операнда, набор арифметических операций с плавающей запятой, группу логических операций и операций управления. Она имела развитую систему прерываний, возможность работы с двойной точностью, высокую скорость реакции на внешние сигналы. ЭВМ М4-2М является синхронной, в ней реализован конвейер команд и конвейер операций и использована необычная разрядная сетка – 29 разрядов. Арифметические операции выполнялись над операндами с плавающей запятой: один разряд был знаковым, 8 разрядов определяли порядок и 20 разрядов составляли мантиссу числа. Таким образом, М4-2М имела расширенный диапазон чисел – порядок от +127 до –128 и уменьшенную мантиссу. Система команд предусматривала операции с получением результата повышенной точности с 40-разрядной мантиссой. В М4-2М за один машинный такт в 4,5 мкс выполнялась любая операция – арифметическая, логическая или управляющая, в

том числе умножение. Это обеспечивалось применением одноконтного АУ с «пирамидой сумматоров» для выполнения умножения. Применена была и другая новинка для тех лет – конвейер.

Запоминающие устройства ОП и ПП строились на ферритовых сердечниках, прошитых в матрице адресными и числовыми обмотками. Позднее в устройстве постоянной памяти матрицы с прошивкой были заменены матрицами с электрической записью, что значительно улучшило технологичность изготовления и позволило вдвое увеличить объем памяти.

Блоки оперативной памяти связаны с устройством управления четырьмя шинами по 29 разрядов, объединявшимися в шину чисел и в шину команд. Таким образом, из устройства ОП могли читаться и операнды и команды. Это было удобно при настройке машины и при отладке рабочих программ пользователей.

Входные регистры арифметического устройства имели непосредственную связь с выходными регистрами, что давало возможность для арифметических операций кроме извлечения операндов из памяти использовать результаты предыдущих операций.



**Структурная схема ЭВМ М4-2М:**

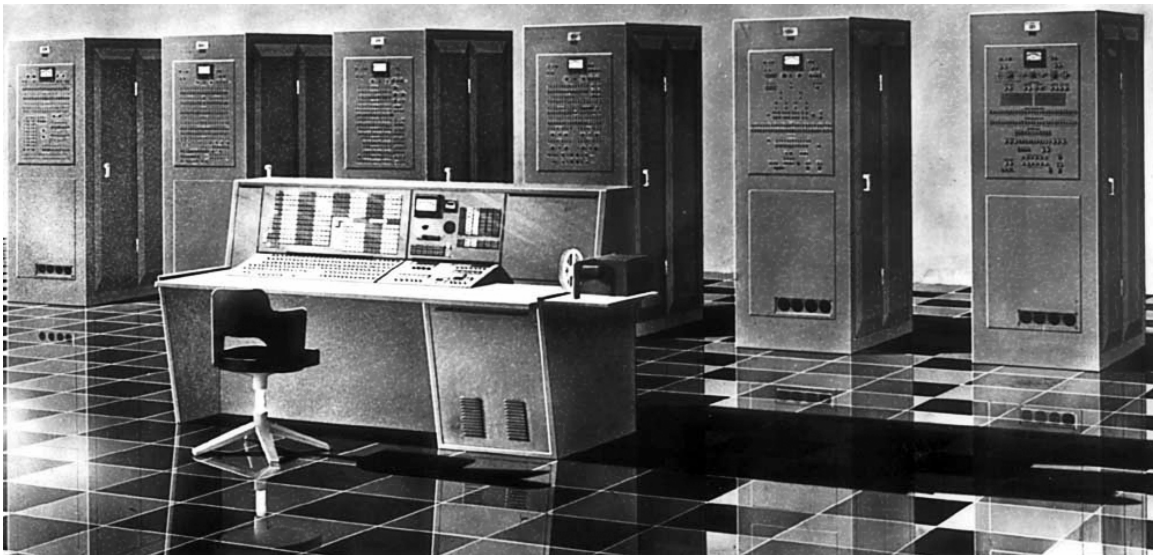
АУ – арифметическое устройство (A, C, B, D – регистры); ОП – оперативная память;  
ПП – постоянная память; ВУ – устройство ввода-вывода

Как управляющая машина ЭВМ М4-2М имела развитую систему внешних прерываний. Обеспечивались 12 активных и 12 пассивных прерываний от объекта управления. Время реакции машины на активные прерывания было очень мало: переход на программу прерывания происходил за два-три машинных такта. Это было важным достоинством ЭВМ М4-2М.

### **Основные технические характеристики**

|                                    |                        |
|------------------------------------|------------------------|
| Система счисления .....            | двоичная               |
| Количество двоичных разрядов ..... | 29                     |
| Представление чисел .....          | с плавающей запятой    |
| Быстродействие .....               | 220 тыс. оп./с.        |
| Объемы памяти модификаций .....    | 5Э71 ... 5Э72 ... 5Э73 |
| оперативная память (Кбайт) .....   | 30 ..... 60 ..... 120  |
| постоянная память* (Кбайт) .....   | 60 ..... 120 ..... 120 |

\* При замене матриц постоянной памяти с прошивкой на матрицы с электрической записью объем памяти ПП в каждой модификации увеличился вдвое.



**Общий вид ЭВМ М4-2М (Мод. 5Э71)**

Машина выпускалась в виде трех модификаций, различающихся объемами оперативной и постоянной внутренней памяти (5Э71, 5Э72, 5Э73). Такие технические характеристики ЭВМ М4-2М позволили использовать ее в построении вычислительных комплексов на разных уровнях системы. С этой целью в дополнение к ЭВМ М4-2М на той же конструктивной и элементной базе были созданы дополнительные системы внешних устройств СВУ-79-1, СВУ-79-2 и внешний вычислитель – ЭВМ М4-3М, которые обеспечили возможность передавать на ГВЦ по каналам связи информацию региональных вычислительных центров, удаленных на тысячи километров.

**Внешний вычислитель** М4-3М работал в единой связке параллельно с машиной М4-2М. Он представляет собой ЭВМ параллельного действия, оперирующую 29-разрядными двоичными числами. Структура командного слова отвечала принципу, принятому для команд ЭВМ М4-2М. Система управления внешнего вычислителя состояла из двух устройств: УВ и ДК. Устройство УВ выполняло классические функции для устройства управления любой ЭВМ: прием и дешифрирование команд, модификацию адреса, выработку управляющих сигналов и т.д. Диспетчер каналов ДК обеспечивал прием и хранение запросов от

различных устройств сопряжения, прием команд от центрального процессора, организацию приоритетного обращения к оперативной памяти, а также выработку управляющих сигналов для устройств сопряжения. Входящие в состав внешнего вычислителя устройства оперативной памяти ОП (такие же, как и в оперативной памяти ЭВМ М4-2М) имели отдельные цепи обращения, благодаря чему одновременно были возможны выборка одного числа, запись информации от устройства сопряжения, выборка чисел по двум адресам и другие совмещенные операции, связанные с обращением к памяти. Арифметическое устройство АВ имело неконвейерную (кольцевую) структуру. Умножение и деление делались «по кольцу»: умножение – за три, деление – за пять тактов. В этом АУ был применен новый алгоритм деления – без восстановления остатка, причем за один цикл вычислялись одновременно четыре бита частного, т.е. деление производилось в шестнадцатеричной системе. Разработку внешнего вычислителя возглавил к.т.н. Ю.Н. Мельник, разработку арифметического устройства – к.т.н. В.А. Брик.

Как отмечалось выше, завод приступил к изготовлению ЭВМ М4-2М в середине 1963 года. В 1964 году была изготовлена первая серия этих машин и началась работа по объединению ЭВМ в вычислительные комплексы на местах их постоянной эксплуатации. С 1965 по 1969 год было построено и введено в эксплуатацию 17 вычислительных комплексов, в том числе пять комплексов с машинами мод. 5Э72 и системами СВУ-79-1 для региональных вычислительных центров и вычислительный комплекс ГВЦ с машинами мод. 5Э73, внешними вычислителями М4-3М и системами СВУ-79-2. В 1969 году территориальная вычислительная система, объединяющая более пятидесяти ЭВМ, была принята в эксплуатацию.

Производство ЭВМ М4-2М продолжалось до 1986 года. Было изготовлено около двухсот таких машин.

### **Третье поколение – ЭВМ М-10 и М-10М**

В 1965–1967 годах в поисках путей повышения производительности вычислительных машин при существующем уровне электроники и технологии Карцев направил основное внимание на исследование архитектур многопроцессорных вычислительных систем. В этих исследованиях он показал четыре вида возможностей распараллеливания вычислений и определил для каждого вида возможную аппаратную реализацию. Особо крупные задачи, решению которых, как правило, и требуется высокая производительность вычислительных средств, обладают в той или иной степени всеми видами параллелизма. Поэтому наиболее универсальный путь создания вычислительных систем с максимально возможной производительностью представляет архитектура многопроцессорной *комбинированной* вычислительной системы. С такой архитектурой в 1967 году был разработан эскизный проект вычислительного комплекса ВК М-9 с производительностью до одного миллиарда операций в секунду. Полностью этот проект реализован не был, но одна из его составных частей – числовая связка – явилась основой машины третьего поколения – ЭВМ М-10.

Техническое задание на разработку ЭВМ М-10 было утверждено в ноябре 1969 года. Перед разработчиками стояла довольно сложная задача: имея микросхемы серии 217 со скоростью срабатывания 15–25 нс, на вентиль и степенью интеграции до 3–5 вентилей в корпусе и ферритовые сердечники М100П2 с внешним диаметром 1 мм нужно было построить ЭВМ с быстродействием не менее 5 млн. оп./с. и с внутренней памятью не менее 5 Мбайт. Для выполнения этой задачи потребовалось 386 тыс. микросхем, 353 тыс. транзисторов и полупроводниковых диодов, 42,5 млн. ферритовых сердечников, 1,4 млн. конденсаторов и 1,4 млн. сопротивлений. Позднее М.А.Карцев написал: «Нельзя сказать, что разработка М-10 была встречена с распростертыми объятьями. Нам говорили, что мы психи, что нельзя заставить работать такую грудку металла, что это все никогда не заработает. Но мы вытянули. Эта работа была отмечена Государственной премией...»

Авторский коллектив разработки: главный конструктор д.т.н. М.А. Карцев, заместители главного конструктора Ю.В. Рогачев, Л.В. Иванов, Л.Я. Миллер, А.А. Крупский, Р.П. Шидловский, основные разработчики В.А. Брик, Е.И. Цибуль, Е.С. Шерихов, Л.З. Либуркин, Г.И. Смирнова, Г.Н. Пусенков, Л.Д. Баранов.

Разработка машины началась в 1970 году. В 1971 году завод получил полный комплект конструкторской документации и приступил к изготовлению машины.

**ЭВМ М-10** – синхронная векторно-параллельная многопроцессорная ЭВМ. Она позволяет использовать практически все виды параллелизма:

- асинхронное выполнение программ разными процессорами с синхронизацией через систему прерываний и обменом информации через канал ввода-вывода (операция типа MIMD);
- синхронное выполнение программ разными процессорами с обменом информацией через систему синхронного обмена (режим «партитуры» программ);
- синхронное выполнение набора операций в одной команде (архитектура «широкой команды»);
- выполнение операции над векторами данных (операция типа SIMD);
- конвейеры команд и операций.

В состав машины входит ряд процессоров различного типа, приспособленных для параллельной работы.

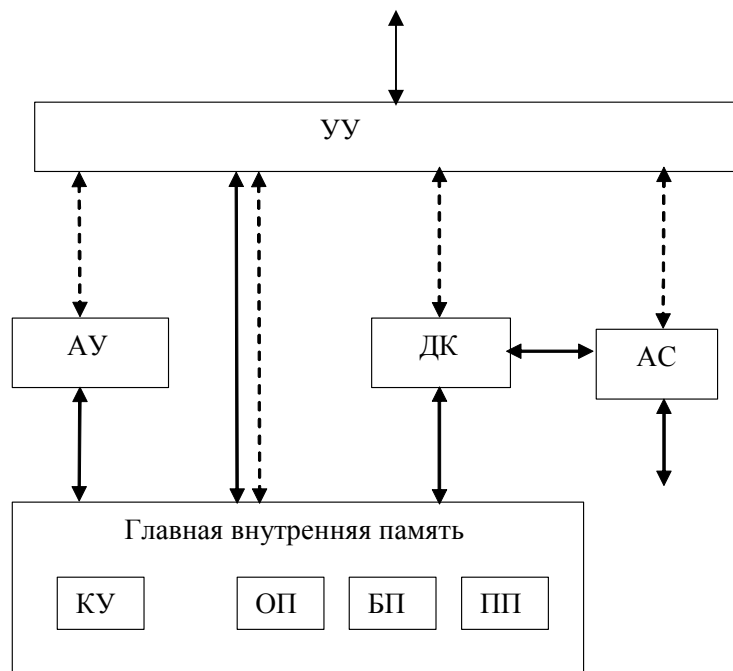
Первый тип – арифметические процессоры, которые физически реализованы в виде двух независимых арифметических устройств, синхронно выполняющих разные арифметические или логические операции. Каждое устройство представлено в виде одного, двух, четырех или восьми процессоров, программно перестраиваемых соответственно в 128-, 64-, 32- или 16-разрядные.

Второй тип процессоров, работающих синхронно с арифметическими, – устройство управления. Функциональная линейка устройства управления реализована с помощью 16-разрядных регистров, названных регистрами признаков. Одновременно с работой этих регистров в устройстве управления выполняется еще полный набор операций с фиксированной запятой над содержимым адресных модификаторов. Таких модификаторов в М-10 шестнадцать. Каждый из них содержит 22 разряда, что соответствует разрядности адреса памяти М-10.

Третий тип процессоров М-10 составляют два синхронно работающих канала связи ПРОЦЕССОР – ПАМЯТЬ, предназначенных для чтения операндов из памяти во входные регистры арифметических процессоров и записи результатов операции в память. Максимальная ширина доступа в память по одному каналу составляет 512 бит, что позволяет заполнять входные регистры всех арифметических процессоров за одно обращение. Внутренняя память машины включает три типа устройств: главная оперативная память объемом 512 Кбайт (ферритовая), постоянная память объемом 512 Кбайт (конденсаторного типа на металлических перфокартах) и большая оперативная память объемом 4 Мбайта (ферритовая). Двусторонний обмен между главной и большой памятью идет со скоростью 20 Мбайт/с в каждую сторону параллельно со счетом в центральном процессоре. Важной особенностью машины является широкий и переменный формат обращения к главной памяти: за одно обращение из нее может быть выбрано от 2 до 64 байтов. Преобразование и коммутация поступающей из ОП и ПП в УУ информации в заданные программой форматы выполняется в коммутационно-кодирующем устройстве (КУ).

Четвертый тип процессора – мультиплексный канал прямого доступа во внутреннюю память (устройство ДК), позволяющий осуществлять ввод-вывод по 24 дуплексным подканалам с суммарной скоростью до 7 Мбайт/с. К каждому подканалу можно подсоединить до шести однотипных устройств.

Следующими параллельными процессорами являются специальные схемы контроля исправности аппаратуры машины и контроля программ пользователей. Взаимодействие этих процессоров осуществляется



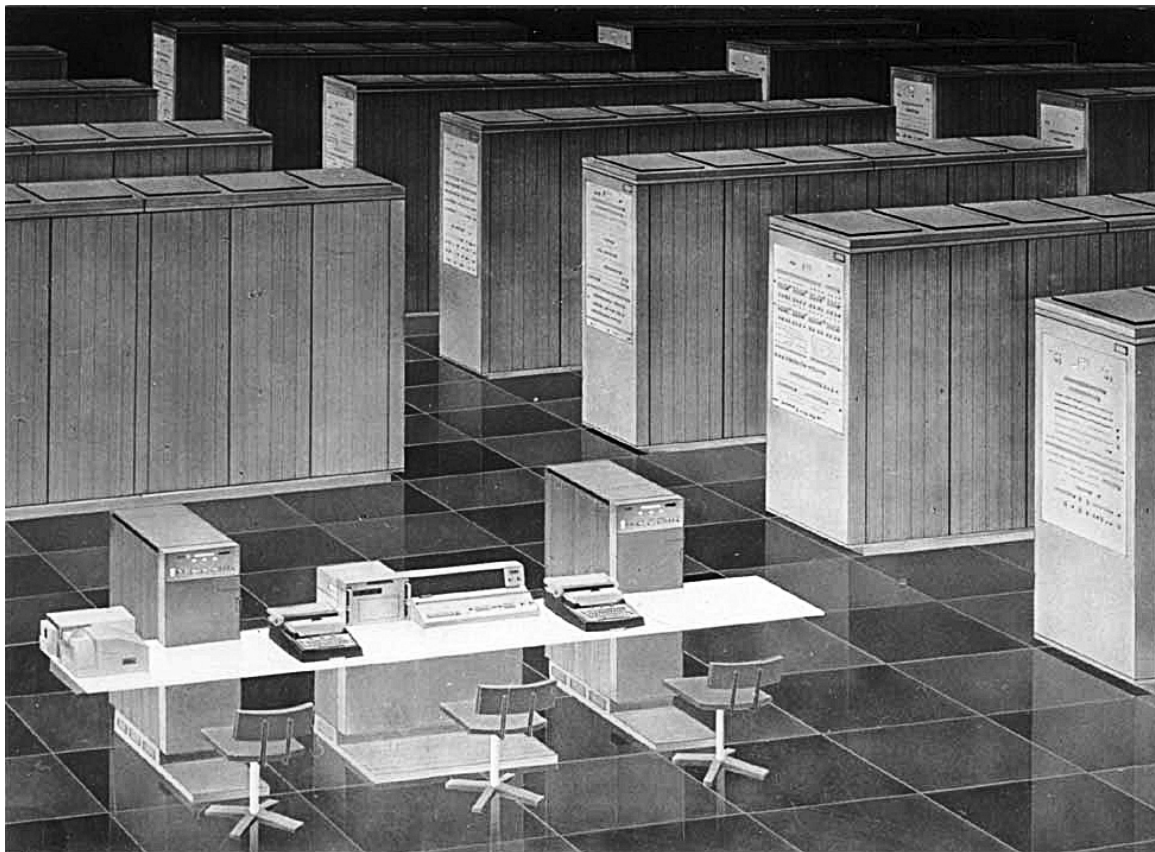
**Блок-схема ЭВМ М-10:**

*УУ* – устройство управления; *АУ* – арифметическое устройство; *ОП* – главная оперативная память; *БП* – большая память (второй уровень оперативной памяти); *ПП* – постоянная память; *КУ* – коммутационно-кодировующее устройство; *ДК* – диспетчер каналов (мультиплексный канал); *АС* – устройство абонентского сопряжения

через многоуровневую систему прерывания программ, являющуюся частью центрального устройства управления. На ее свободные входы могут приниматься также внешние сигналы (до 32).

В машине предусмотрены цепи, позволяющие объединять до семи машин М-10 в единый синхронный комплекс, работающий от общего тактового генератора. В каждом такте машина, работающая в комплексе, может выдать на свои выходные шины массив данных в 64 байта и принять массив такого же размера от любой другой машины.

Все оборудование ЭВМ М-10 размещалось в 31 типовом шкафу (на фотографии показана их вдвоенная установка), из них 21 шкаф занимало оборудование запоминающих устройств. К 1974 году было изготовлено три комплекта машин для построения вычислительного комплекса. Одновременно с работами по объединению машин в единый комплекс на этих же машинах выполнялась отладка внутреннего программного обеспечения машины и пользовательских программ комплекса. В 1976 году ЭВМ М-10 и вычислительный комплекс успешно выдержали государственные испытания. Вычислительный комплекс был введен в штатную эксплуатацию в режиме непрерывной круглосуточной работы. Производство машин М-10 продолжалось. В то же время были разработаны новые устройства памяти с использованием появившихся больших интегральных схем, которые в 2,5 раза сократили количество шкафов. Машина с новыми устройствами получила обозначение М-10М, и с 1977 года в производстве шла машина уже в новой комплектации. Вычислительные машины М-10 и М-10М программно совместимы и полностью взаимозаменяемы. На их основе создан целый ряд уникальных вычислительных комплексов, включая и вычислительные комплексы, объединяющие шесть машин. Некоторые вычислительные комплексы работают до настоящего времени. Производство этих машин продолжалось до 1992 года, было выпущено более 50 экземпляров.



*Общий вид ЭВМ М-10*

#### **Основные технические характеристики**

Среднее быстродействие .....5,1 млн. оп./с  
Общий объем внутренней памяти .....5 Мбайт  
Емкость буферной памяти мультиплексного канала .....64 Кбайта  
Система прерывания программ .....72-канальная, 5 уровней приоритетов

Обеспечивается одновременная работа в режиме разделения времени 8 пользователей на 8 математических пультах. За один машинный такт одновременно выполняются операции с фиксированной и плавающей запятыми. Предусмотрены векторные операции. Например, за один такт может быть произведено вычисление скалярного произведения векторов.

Математическое обеспечение ЭВМ М-10 включает:

- операционную систему, обеспечивающую разделение времени и оборудования, диалоговый режим одновременной отладки до 8 независимых программ и мультипрограммный режим автоматического прохождения до восьми независимых задач;
- систему программирования, включающую машинно-ориентированный язык АВТОКОД, проблемно-ориентированный язык АЛГОЛ-60 и соответствующие трансляторы и средства отладки;
- библиотеку типовых и стандартных программ;
- диагностические программы;
- программы контроля функционирования.

По мере расширения сферы применения машин у различных пользователей состав математического обеспечения пополнялся новыми программами.

Оценивая вклад архитектуры ЭВМ М-10 в развитие вычислительной техники в СССР, профессор Б.А. Головкин писал:

«...М.А. Карцевым предложена, насколько можно судить – впервые в мире, концепция полностью параллельной вычислительной системы с распараллеливанием на всех четырех уровнях (программ, команд, данных и слов) и, что очень важно, эта концепция реализована в виде созданных на базе ЭВМ М-10 вычислительных комплексов.

Вклад параллельной архитектуры в повышение производительности оказался столь весомым, что на большой длительности машинного такта в 1,9 мкс (из-за несовершенной элементной базы) производительность ЭВМ М-10 на госиспытаниях оказалась 5,1 млн. оп./с. (в пиках – значительно выше). ЭВМ М-10 вплоть до развертывания МБК «Эльбрус» оставалась наиболее мощной отечественной ЭВМ.

Таким образом, по архитектурным принципам, по концепции и реализации ЭВМ М-10 опередила собой новый класс машин и была первой в стране реализацией векторно-параллельных ЭВМ...

#### **Четвертое поколение машин М.А. Карцева – ЭВМ М-13**

К концу 1977 года уже стало ясно, что идея многопроцессорных вычислительных машин стала реальностью, а машины М-10 подтвердили ее широкие возможности. Работая над проектом вычислительной машины нового, четвертого поколения, М.А. Карцев опирался на опыт создания ЭВМ М-10 и вычислительных комплексов на ее основе. Этот опыт показывал, что структура новой машины должна быть более гибкой по производительности и по комплексированию. Такую задачу поставил Карцев перед коллективом, приступая к разработке новой многопроцессорной вычислительной машины М-13.

Проект ЭВМ М-13 предусматривал три базовые модели, а также ряд их модификаций, различающихся комплектностью арифметических устройств, устройств памяти, дополнительных внешних устройств и др. Все модели строились по модульному принципу, используя одну и ту же номенклатуру устройств. Переход от малой модели к средней и максимальной производился путем увеличения количества модулей.

Программная совместимость систем М-13 обуславливалась единым для всех моделей машинным языком и единым математическим обеспечением, содержащим режим работы в реальном масштабе времени, диалоговый режим разделения времени с предоставлением мониторов для создания, трансляции и отладки программ на машинно-ориентированных (АВТОКОД М-13), проблемно-ориентированных (АЛГОЛ-60, ФОРТРАН, КОБОЛ) и универсальных (АЛГОЛ-68) языках.

В части элементной, конструктивной и технологической базы М-13 использовались наиболее прогрессивные технические решения. В логических узлах применялись микросхемы типа ТТЛ серий 133, 130, 530. Для запоминающих устройств использовались микросхемы полупроводниковой памяти широкого применения.

Техническое задание на разработку ЭВМ М-13 и вычислительного комплекса на ее основе было утверждено в 1980 году. Главным конструктором был утвержден М.А. Карцев, заместителями главного конструктора – Ю.В. Рогачев, Л.Я. Миллер, А.Ю. Карасик, Л.В. Иванов, Р.П. Шидловский, А.А. Крупский, Е.И. Цибуль. Разработка конструкторской документации была завершена в 1982 году.

М-13 является многопроцессорной векторно-конвейерной ЭВМ с конвейеризацией на уровне данных. В ней имеется векторное арифметическое устройство с параллельными перестраиваемыми конвейерными устройствами обработки, каждое из которых выполняет в данный момент в режиме конвейера одну и ту же операцию, но над различными операндами. В состав М-13 входит центральная процессорная часть, аппа-

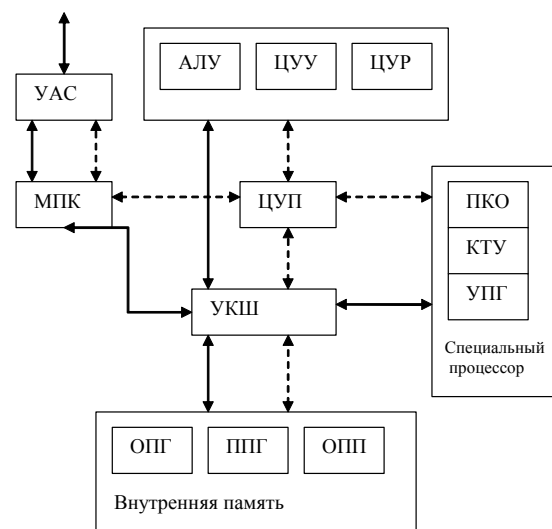
ратные средства поддержки операционной системы, подсистема ввода-вывода и специализированная процессорная часть.

**Центральная процессорная часть**, предназначенная для основных вычислений, содержит арифметические процессоры (АЛУ), устройства внутренней памяти (ОПГ, ППГ, ОПП), центральное устройство управления (ЦУУ) и устройство редактирования (ЦУР).

АЛУ – векторное конвейерное арифметико-логическое устройство. В зависимости от формата операндов (8, 16 или 32 двоичных разряда) распадается соответственно на 16, 8 или 4 арифметических процессора. ЭВМ М-13 могла содержать 1, 2 или 4 АЛУ, которые в любом сочетании составляют единое векторное устройство. Все арифметические процессоры всех АЛУ работают синхронно, выполняя одну и ту же операцию над всеми парами операндов. Каждый арифметический процессор представляет собой конвейерное устройство. АЛУ оперирует числами как с фиксированной, так и с плавающей запятой.

ЦУУ – центральное устройство управления содержит булевский процессор для управления потоками команд и для маскирования при векторной обработке, а также процессор адресной модификации с производительностью 3 млн. оп./с для управления адресным пространством.

ЦУР – центральное устройство редактирования – обеспечивает выполнение целого ряда специальных процедур, связанных с взаимодействием линейки процессоров и блоков памяти. К ним относятся: уплотнение массивов под маской с целью исключения пробелов, которые появляются в процессе параллельной обработки; кольцевой сдвиг строки информации; транспозиция строки информации. Кроме этого в устройстве реализованы процедуры, связанные с организацией вычислительного конвейера: трехадресное обращение к регистровой памяти (2 чтения + 1 запись одновременно) для ускорения обслуживания АЛУ и согласование форматов обработки в разных комплектациях ЭВМ М-13 с сохранением их программной совместимости.



**Блок-схема ЭВМ М-13**

**Аппаратные средства поддержки операционной системы** включают центральный управляющий процессор (ЦУП) и устройство управления кодовыми шинами (УКШ).

ЦУП – центральный управляющий процессор – обеспечивает аппаратную поддержку операционной системы, а также аппаратную поддержку виртуальной памяти и управления процессами. Связан управляющими интерфейсами со всеми процессорами машины.

УКШ – устройство управления кодовыми шинами – содержит таблицы виртуальной (математической) памяти. Связывает широкоформатными шинами все устройства машины с внутренней памятью.

Подсистема ввода-вывода включает мультиплексный канал (МПК) и устройство абонентского сопряжения (УАС).

МПК – мультиплексный канал – предназначен для сопряжения подсистемы ввода-вывода с центральным управляющим процессором и внутренней памятью. Реализует запуск обменных операций в каналах, диспетчеризацию работы каналов, организацию виртуальной и относительной адресации и обработку прерываний от каналов. Выполняет также мультиплексирование обмена данными между устройствами абонентского сопряжения и внутренней памятью и осуществляет преобразование форматов при обмене данными.

УАС – устройство абонентского сопряжения – содержит сопрягающие процессоры, которые позволяют подключать к машине как стандартные, так и специализированные устройства, входящие в состав управляемых объектов. В зависимости от комплектации в УАС могло входить от 4 до 128 одинаковых сопрягающих процессоров с программируемым абонентским интерфейсом, что обеспечивало решение проблемы сопряжения машины с разнообразными и уникальными интерфейсами абонентов реального времени.

**Специализированная процессорная часть** содержит процессоры когерентной обработки (ПКО), контроллер технического управления (КТУ) и управляющую память гипотез (УПГ).

ПКО – процессор когерентной обработки – представляет собой специализированный векторно-конвейерный вычислитель. В нем используется программно-управляемая глубококонвейерная архитектура устройства двухточечного преобразования, основу которого определяет узел для выполнения базовой операции быстрого преобразования Фурье. Применение этой базовой операции позволило на том же оборудовании выполнять многие другие операции, необходимые в алгоритмах цифровой обработки сигналов: вычисление максимального значения в массиве, сравнение массива с пороговым значением, вычисление суммы произведения массивов, вычисление корреляционных матриц и др. ПКО производит аппаратно умножение двух комплексных чисел. В одном шкафу 4 процессора. Эквивалентная производительность одного шкафа 120 млн. оп./с. Допускается комплектация от 1 до 20 шкафов.

КТУ – контроллер технического управления – предназначен для сопряжения специализированной (СПЧ) и центральной (ЦПЧ) процессорных частей, а также для диспетчеризации различных групп ПКО. Информационный тракт обмена с внутренней памятью центрального процессора реализован с помощью шин чтения и шин записи. КТУ выполняет в мультиплексном режиме обмен данными по вертикальному интерфейсу между отдельными группами ПКО внутри СПЧ (межгрупповой обмен), а также между внутренней памятью ЦПЧ и отдельными группами ПКО. При этом производится согласование форматов и преобразование представления чисел из формата СПЧ в формат ЦПЧ.

УПГ – устройство памяти гипотез – специализированное многопортовое запоминающее устройство. Используется в СПЧ для «длительного» хранения «системной» информации ПКО и для «временного» хранения входной, выходной и промежуточной (рабочей) информации.

Серийное производство машины М-13 началось в 1984 году, с задержкой на два года, уже после смерти М.А. Карцева. Автору этой статьи, ставшему его преемником, удалось добиться решения о включении в план Загорского электромеханического завода на 1984 год начало изготовления машин М-13 средней комплектации. В 1986 году завод поставил первые две машины, и в дальнейшем их изготовление шло строго по плану. В 1991 году вычислительный комплекс из шести ЭВМ М-13 успешно выдержал государственные испытания и был принят в штатную эксплуатацию для работы в режиме круглосуточной непрерывной работы в реальном масштабе времени.



Общий вид ЭВМ М-13

**Технические характеристики ЭВМ М-13**

## Центральная процессорная часть

Быстродействие, оп./с ..... 12 млн. .... 24 млн. . 48 млн.

Емкость внутренней памяти, Мбайт ..... 8,5 ..... 17 ..... 34

В том числе:

ОПГ (1-й уровень), Мбайт ..... 0,25 ..... 0,5 ..... 1,0

ППГ, Мбайт ..... 0,25 ..... 0,5 ..... 1,0

ОПП (2-й уровень), Мбайт ..... 8,0 ..... 16 ..... 32

Суммарная пропускная способность

центрального коммутатора, Мбайт/с ..... 800 ..... 1600 ..... 3200

Пропускная способность

мультиплексного канала, Мбайт/с ..... 40 ..... 70 ..... 100

Абонентское сопряжение:

Число сопрягающих процессоров ..... 8 ..... 16 ..... 128

Максимальное быстродействие, оп./с ..... 350 млн.

## Специализированная процессорная часть

Емкость управляющей памяти гипотез, Мбайт ..... 4 ..... 8 ..... 128

Максимальное эквивалентное быстродействие, оп./с ..... 2,4 млрд.

## Внешняя память

На сменных магнитных дисках, Мбайт ..... 200

На магнитной ленте, Мбайт ..... 42

Вычислительные машины комплекса на испытаниях показали следующие данные:

- производительность ЦПЧ: 24 млн. операций в секунду,
- объем внутренней памяти ЦПЧ: 17 Мбайт,
- производительность СПЧ: 1,2 млрд. операций в секунду,
- объем внутренней памяти СВЧ: 40 Мбайт.

Производство машин продолжалось до 1992 года. Было изготовлено 18 комплектов этих машин.

## ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА В КАРЕЛИИ

**Н.С. Рузанова**

Петрозаводский государственный университет

Петрозаводск, РФ

*ruzanova@psu.karelia.ru*

История развития ВТ в Карелии началась в 1960 году, когда в Петрозаводском университете была установлена первая вычислительная машина Минск-1. В 1961 году она была введена в эксплуатацию, и с этого года ВЦ стал учебной базой для подготовки студентов физико-математического факультета в области ВТ и программирования. Уже в следующем году на базе ВЦ началась подготовка операторов-программистов.

В 1961 году ЭВМ «Минск-1» была установлена в отделе энергетики и автоматики Карельского филиала АН СССР (КФАН), что положило начало использованию ЭВМ в научных исследованиях. В 1965 году «Минск-1» была введена в эксплуатацию в Карельском научно-исследовательском институте лесной промышленности (КарНИИЛП), куда в результате реорганизации влилась большая часть программистов КФАН. В 1967 году в КарНИИЛПе была установлена ЭВМ «Минск-22». Это позволило начать работы по созданию АСУ Кареллеспрома, включающего и такие задачи, как «Учет лесного фонда» и «Учет лесосечного фонда».

В 70-е годы на крупных предприятиях и в организациях Карелии устанавливалась, в основном, ЭВМ «Минск-32». Так, в 1971 году эту ЭВМ получили Главсевзапстрой и завод бумагоделательного и химического машиностроения (ныне Петрозаводскмаш), в 1973 – Беломорско-Онежское пароходство, в 1975 – Статистическое управление КАССР, в 1977 – Онежский тракторный завод и другие. Это были годы создания ВЦ, основной задачей которых была разработка и внедрение автоматизированных систем управления предприятиями (АСУП). А в конце 70-х осуществлялся массовый переход ИВЦ на машины серии ЕС. Практически во всех организациях на смену Минск-32 пришла ЭВМ ЕС 1022. Получили эту ЭВМ и вновь созданные ИВЦ в Карелэнерго, Карелгражданпроекте, Слюдяной фабрике и др. В начале 80-х годов устанавливались более мощные ЕС ЭВМ. Так, КарНИИЛП вводил в эксплуатацию ЕС 1032, ЕС 1045, ЕС 1061; Главсевзапстрой – ЕС 1045; завод «Петрозаводскмаш» – ЕС 1055; Онежский тракторный завод – ЕС 1045; Радиозавод – ЕС 1033; Статуправление – ЕС 1035, ЕС 1036; в КФАН была введена в эксплуатацию ЕС 1052. Во второй половине 80-х годов в организациях и предприятиях использовались мини- и микро ЭВМ (СМ 4, СМ 1600, Электроника 60, ДВК и другие), в основном для инженерных расчетов, а также для создания АСУТП.

Программистов для ВЦ Карелии готовил Петрозаводский университет, на ВЦ которого также происходила смена техники: ЭВМ «Одра 1204» (1970), «Минск 22» (1972), ЕС 1022 (1978). В 1985 году в университете была установлена ЕС 1035, укомплектованная дисплейной станцией, позволяющей подключать до 32 дисплеев. Это дало возможность организовать дистанционный доступ пользователей к ресурсам ЭВМ. Параллельно с начала 80-х годов в университете эффективно использовались СМ 4, Мера 125, а также микроЭВМ Электроника 85, Электроника 60, Ямаха, ДВК, ЕС 1840, Искра 1030, Mazovia 1016, Macintosh и другие. Начиная с 90-х годов, университет готовит студентов на самых современных персональных компьютерах IBM PC/XT, PC/AT (1990-е годы), P1, PII, PIII, PIV (2000–06 гг.)

С 1994 года созданный на базе ПетрГУ Региональный Центр Информационных технологий начал предоставлять услуги доступа в Интернет. В 1996 году была принята и утверждена Правительством РК первая Программа «Информатизация Республики Карелия на 1996–1998 гг.», подготовленная совместно ПетрГУ и Карельским Научным Центром РАН, которая заложила основы стратегического подхода к развитию информатизации в Карелии и стимулировала использование IT-технологий во всех сферах деятельности в республике.

В 2000–2005 гг. на основе сетевых технологий активно внедрялись АСУП, средства автоматизации документооборота, автоматизированного проектирования, осваивались технологии реализации ГИС-систем.

## ШКОЛА АКАДЕМИКА С.А. ЛЕБЕДЕВА. ПЕРВЫЕ ШАГИ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ ЭВМ

**Г.Г. Рябов**

Институт точной механики и вычислительной техники РАН  
Москва, РФ  
*direction@ipmce.ru*

В конце 50-х годов, уже после создания машин БЭСМ и БЭСМ-2, академик С.А. Лебедев в своих рабочих тетрадях сделал набросок ряда важных задач в проектировании будущих ЭВМ, которые невозможно было решить без применения самих ЭВМ.

К таким задачам были отнесены: моделирование общей логической структуры проектируемой ЭВМ как особой системы массового обслуживания, эмуляция системы команд проектируемой машины на существующих ЭВМ и ряд технических задач. Причем С.А. Лебедевым не только были поставлены задачи, но и указаны возможные методы их решения.

Так, им была предложена статистическая модель программы, при прогоне которой с ориентировочными временами выполнения отдельных операций можно было получить оценки производительности проектируемой машины. Впоследствии близкая методика была с успехом использована при создании легендарной БЭСМ-6.

С большим пониманием относился С.А. Лебедев к роли дискретной математики в области технического проектирования ЭВМ. При переходе к разработке машин на интегральных схемах и печатных платах академик С.А. Лебедев как главный конструктор являл собой пример глубокого аналитика при выборе научно обоснованных конструктивных решений. В период создания вычислительного комплекса реального времени 5Э26 он не побоялся в разгар выполнения проекта внести существенные коррективы в конструкцию, когда стал возможен математический прогноз некоторых параметров конструкции.

Такое положение во многом определило атмосферу и усилия, которые были направлены на разработку в ИТМ и ВТ сквозной системы автоматизированного проектирования, которая стала затем базой при создании суперЭВМ «Эльбрус».

## ОЧЕРК ИСТОРИИ СТАНОВЛЕНИЯ И РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ В РЕСПУБЛИКЕ БЕЛАРУСЬ

**О.И. Семенков**

ОИПИ НАН Беларуси  
Минск, Беларусь  
*semenkov@basnet.by*

После восстановления разрушенного войной промышленно-технологического и научного потенциала Белоруссия на карте СССР приобрела устойчивый имидж машиностроительно-аграрной республики. Однако с конца 50-х годов в результате принятых на государственном уровне принципиальных решений в рамках общесоюзной координации и разделения труда, а также благодаря усилиям тогдашнего руководства республики этот имидж начал претерпевать серьезные изменения. В Белоруссии получили развитие исследования в области точных наук, на существовавших к тому времени предприятиях приборостроения и радиотехнической промышленности началось производство в широких масштабах наукоемкой продукции,

высшая школа обеспечила подготовку высококвалифицированных специалистов в области новых и высоких технологий. Все это послужило базой для создания и последующего интенсивного развития в республике отраслей микроэлектронной промышленности и промышленности средств вычислительной техники. Невозможно переоценить всестороннюю помощь, которую оказала в то время Россия, направив своих высококвалифицированных инженеров и специалистов в новые отрасли промышленности, обеспечив подготовку молодых ученых для развития необходимых научных исследований и разработок в Академии наук и отраслевых НИИ и КБ.

В докладе приводится хронология создания предприятий микроэлектроники и вычислительной техники, объемы производства и тактико-технические характеристики продукции, выпускавшейся в Белоруссии на протяжении 60–80-х годов.

В это же время создаются получившие в последующем широкую известность научные школы в области вычислительной математики, информатики, систем автоматизации научных исследований, проектирования и управления, обработки изображений и цифровой картографии. Дается анализ и оценка уровня достижений этих школ и их вклада в создание новых информационных технологий.

Качественно новым этапом в истории информатики и вычислительной техники явилось выполнение совместных белорусско-российских научно-технических программ по созданию семейства суперкомпьютеров и прикладных систем на их основе в рамках Союзного государства Белоруссии и России. Обсуждаются результаты реализации этих программ, приводятся характеристики и тактико-технические данные созданных моделей семейства суперкомпьютеров «СКИФ» и обсуждаются пути их использования в научных исследованиях и промышленности. Даются прогнозные оценки на ближайшую перспективу.

## ФОНДОВОЕ СОБРАНИЕ «ЭЛЕКТРОННЫЕ ЦИФРОВЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ» В ПОЛИТЕХНИЧЕСКОМ МУЗЕЕ

**М.Э. Смолевицкая**

Политехнический музей

Москва, РФ

*smol@polymus.ru*

Счетные приборы и вычислительные машины Политехнический музей начал собирать еще в первые годы своего существования, т.е. с начала 1870-х годов. Сейчас музей располагает фундаментальным собранием вычислительной техники, которое состоит из 7 систематических коллекций и 11 персональных фондов ученых.

Фондовое собрание электронных цифровых вычислительных машин (ЭЦВМ) формируется в музее относительно недавно, с 1960-х годов. В данный момент в нем насчитывается 250 музейных предметов и более 1 600 единиц хранения документальных материалов. Образовано 11 личных фондов ученых, посвятивших свою деятельность электронной вычислительной технике. При этом важно отметить, что данное фондовое собрание является единственным в нашей стране.

Мы знакомим наших посетителей с памятниками науки и техники: ЭВМ «Урал-1», единственной в мире ЭВМ на троичной логике «Сетунь». Только у нас в музее можно увидеть электронно-лучевую трубку – элемент оперативной памяти первой серийной отечественной ЭВМ «Стрела», различные варианты памяти на ферритовых сердечниках, постоянное запоминающее устройство конденсаторного типа на металлических

перфокартах ЭВМ М-10, которая разрабатывалась под руководством М.А. Карцева. Среди поступлений последних лет – вычислительная система «Эльбрус 3.1», бортовой вычислительный комплекс космического назначения «Аргон-16», подобный работавшему на орбитальной космической станции «Мир».

Хронология достижений и вклад отечественных компьютерных архитекторов должны быть известны широкой общественности.

Экспозиционный комплекс «Основоположники отечественной вычислительной техники» позволяет более полно отразить не только историю развития электронной вычислительной техники, но и показать вклад отечественных ученых в развитие теории и практики отечественного и мирового компьютеростроения и информатизации общества. В этой экспозиции представлены научные труды, копии дипломов и бронзовых медалей, свидетельствующих о присвоении звания (IEEE Computer Society) «Computer Pioneer» С.А. Лебедеву, А.А. Ляпунову и В.М. Глушкову, что позволяет продемонстрировать международное признание заслуг наших ученых в этой области.

С 1994 года Отделение информатики, вычислительной техники и автоматизации (ОИВТА) Российской академии наук один раз в два года присуждает премию имени С.А. Лебедева за достижения в области разработки вычислительных систем. Музей хранит копии дипломов Президиума РАН о присуждении отечественным ученым этой премии.

## DEVELOPMENT OF COMPUTER SCIENCE IN SIBERIA

**Ya.I. Fet**

Institute of Computing Mathematics and Mathematical Geophysics of Siberian Branch of Russian Academy of Sciences

Novosibirsk, Russia

*fet@ssd.ssc.ru*

### 1. Introduction

The establishing of the Siberian Branch of USSR Academy of sciences as well as the creative work of scientists of this new scientific center during the following years was of great importance to the history of computer science in Russia (Soviet Union). Looking at these events in detail one can realize a principal contribution of Siberian scientists to the automata theory, theory of computing systems, parallel and distributed computing technique, theoretical and applied programming, etc.

Each of these subdivisions was represented by a definite scientific school, with a distinguished scientific leader at the head. Together with his students and followers, the leader carried out investigations in the corresponding field.

It is well known that in the '60s the Siberian Branch attracted great attention of the world scientific community. It was just because leading Soviet mathematicians – Mikhail Lavrentiev, Sergey Sobolev, Leonid Kantorovich, Guriy Marchuk, Alexey Lyapunov, Andrey Ershov and others moved to Novosibirsk. In the Siberian Branch, from the very beginning, priority was given to the development of computational mathematics, mathematical modeling, computer technology, programming. The latest models of domestic computers have been working here.

The founders of the Siberian Branch believed that all sciences (even the Humanities) can be sufficiently advanced if the newest achievements of the computer science will be mastered. In the very beginning of the Siberian Branch, at the end of '50s, Lavrentiev wrote: «Instead of building and testing the real installations, we can now estimate, by using computers, every version of the process changing arbitrary its parameters».

Thus, we become firmly convinced that during the '60s and '70s years of the last century the point of concentration of cybernetic research moved to Novosibirsk.

There can be no doubt that successful development of computer science in Siberia is connected with the personality of academician Lavrentiev. He is known as one of initiators of the Siberian Branch, as its President. But one should not forget that it was exactly Lavrentiev who saw Computer Science and Computer Technology in a proper perspective as early as in the end of '40s. Still working in Kiev, Lavrentiev sent a letter to Stalin emphasizing the importance of support and development of computer technology. It was Lavrentiev who headed during some period the Institute ITMVT and invited to this Institute Sergey Lebedev. We can assert that thanks to Lavrentiev ITMVT become the origin and glory of Soviet computer technology.

In this report, we intend to present some selected pages from the history of Computer Science in Siberia.

## **2. Alexey Lyapunov**

In the history of Russian science, Professor Alexey Lyapunov (1911–1973) occupies a particular place in connection with his activities in defending cybernetics and genetics.

A.A. Lyapunov was a representative of the old Russian nobility which originated numerous distinguished workers of Russian science and culture during 19-th and 20-th centuries. While still a young talented mathematician who already was a hero of the Great Patriotic War and a Professor of the Department of Mathematics at the Moscow Artillery Academy, Alexey Lyapunov headed, in the early '50s, the struggle for the recognition of cybernetics in the USSR. In those hard times, when dogmatic philosophers denounced cybernetics as a «reactionary pseudo-science», Professor Lyapunov, in the 1954/1955 academic year organized a Seminar on cybernetics at the Moscow State University. This Seminar attracted, from the very beginning, great attention of different specialists and developed into an all-Moscow and even in an all-Union event. He invited for work in this Seminar and presenting lectures the youth as well as prominent Soviet scientists, and sometimes even foreign guests. During ten years (1954–1964) a total of 121 sessions of this famous «Big» Lyapunov's Seminar was held.

In 1958, he launched his famous series of collections «Problemy Kibernetiki» («Systems Analysis») being the editor-in-chief of it. The total of 41 issues have been published during the period 1958–1984.

The fearless struggle for the new science was crowned with full victory.

Lyapunov was a distinguished mathematician, a specialist in the theory of sets. However, the range of his interests was so widespread that he can be rightfully named a person of encyclopedic knowledge. The known Russian historian of science Modest Haase-Rapoport wrote about Alexey Lyapunov: «Despite the broad spectrum of his scientific interests, Lyapunov's activities in science were always distinguished by professional skill. The biologists considered him a biologist, the geophysicists a geophysicist, the philosophers a philosopher» [1]S.

Beginning from 1961 Professor Lyapunov lived in Novosibirsk and worked at the Siberian Division of Russian Academy of Sciences. This was one of the most fruitful periods of his life. Alexey Lyapunov was an outstanding pedagogue, and a propagandist of scientific knowledge. His interests in this field covered the teaching at all the levels of education, from the primary school to the University.

Dr. Igor Poletayev, one of the Russian Computer Pioneers and close Lyapunov's collaborator wrote: «Scientific truth ever was the subject of his service, and the search of this truth – almost a cult. This unselfish, knightly service to the truth was supplemented by his irresistible personal charm <...> and surprisingly deep and, perhaps, inoffensive and kind humor. Even the controversial opinions sounded from Lyapunov's lips attractive, almost convincing. Every conversation and personal contact with him was an intellectual event and aesthetic experience».

A.A. Lyapunov was a distinguished teacher and disseminator of scientific knowledge. His interests covered the teaching at all the stages of education, from the university down to the primary school level. Lyapunov's pedagogic activity reached its highest point at the Novosibirsk Academic Village. Together with M.A. Lavrentiev, he initiated the establishment at the Novosibirsk University of the first «Physico-Mathematical School» in Russia. He also was one of the organizers of Siberian Mathematical Olympiads and Summer Physico-Mathematical Schools.

Aleksey Lyapunov is one of the two outstanding Russian computer scientists honored in 1996 as Computer Pioneers. The IEEE Computer Society stated that «Alexey Lyapunov... founded Soviet cybernetics and programming». The second Laureate was Sergey Lebedev (1902–1974), the «founder of Soviet computer industry».

It is known that the IFIP WG 9.7 (History of Computing) established a Program of «Pioneer Days» to recognize contribution of distinguished scientists awarded as Computer Pioneers.

Between October 8 and 11, 2001 an International Conference devoted to the 90-th anniversary of Lyapunov's birth was held in Novosibirsk. Within the framework of the Conference a Pioneer Day was organized on October 8, 2001 to celebrate Aleksey Lyapunov's memory. This was the first such event devoted to a Russian Computer Pioneer. At the beginning of the Pioneer Day Session, Dr. Gyoza Kovacs (Budapest) made a brief introduction, describing the international computer community's tradition of celebrating Computer Pioneers and previous Pioneer Days. Then, Natalya Lyapunova, the scientist's daughter, presented some reminiscences of her father. She showed the audience Lyapunov's «Computer Pioneer» medal as well as some IEEE Computer Society documents related to this Award.

### 3. Leonid Kantorovich

Leonid Kantorovich (1912–1986) is one of those brilliant scientists whose lives are inalienable from the 20th century history of Russia. One might say that Kantorovich lived two different lives in science, as a mathematician and as an economist. Kantorovich-mathematician started his studies at the age of fifteen and soon was acknowledged as a leading scientist.

The name of Leonid Kantorovich, his life, his role in the science, and his struggle for his ideas occupy a special position in the history of science of the 20th century.

Kantorovich was born in St. Petersburg, in a family of a physician. He showed his talent very early. He entered the Leningrad University in 1926, being only fourteen. Having graduated in 1930, Kantorovich engaged in teaching activity and active scientific work. In 1932 he was appointed a professor at the Leningrad Institute of Engineers of Industrial Construction and a senior lecturer at the Leningrad University, his alma mater, where he took professorship in 1934 and the degree of a Doctor was conferred on him without a thesis defense in 1935. Kantorovich remained associated with the Leningrad University.

The early blossoming of his talent, the discovery (at the age of 27) of new methods of planning and management, the extraordinary breadth of interests, the uncompromising nature of a fighter, and, at the same time, his modesty and nobility – all these features form the unique phenomenon of Kantorovich.

The mathematical investigations of Kantorovich formed a basis of new important directions in mathematics. At the same time, he is rightfully considered one of the founders of modern mathematical economics, the kernel of which is linear programming which he created. It is a new concept of the economical cybernetics which is highly important because it allows to transform economics into an objective science, thus ensuring most efficient results of economic activity.

Functional analysis plays a special role in Kantorovich's mathematical activity. Being a classic in this area and an author of the theory of ordered vector spaces, Kantorovich made functional analysis a natural language of computational mathematics

He realized the importance of computer technology at its very dawn and believed that this invention «will influence all human activities as greatly as did book-printing, steam engine, electricity, and radio».

Kantorovich's famous book «The Best Use of Economic Resources» published in 1959 faced hostile criticism from Soviet economists and caused much debate which lasted as long as until the mid-1960s, attracting keen interest of foreign scientists. It was the time when his first works on linear programming, translated into foreign languages, became famous and brought him priority and recognition in the West.

Kantorovich is the author of more than 300 scientific works, which encompass an impressively broad range of fields, but are united by his personality as well as by the inherent integrity and mutual association of ideas.

The name of Leonid Kantorovich, his life, his role in the science, and his struggle for his ideas occupy a special position in the history of science of the 20th century. The mathematical investigations of Kantorovich formed a basis of new important directions in mathematics. At the same time, he is rightfully considered one of the founders of modern mathematical economics, the kernel of which is linear programming which he created. It is a new concept of the economical cybernetics which is highly important because it allows to transform economics into an objective science, thus ensuring most efficient results of economic activity.

In 1939 Leonid Kantorovich published a small brochure «Mathematical Methods in the Organization and Planning of Production» containing the discovery of linear programming, a new scientific direction which had an enormous influence on the development of economic science. This fundamental work of Kantorovich gave, for the first time, a mathematical statement of economic problems of planning and suggested efficient methods of their solution.

The new, truly scientific approach to economics suggested by L. Kantorovich was incompatible with the official Soviet ideology. For a long time, Kantorovich's discovery of linear programming had remained unknown to the West. At the end of the '40s, similar approaches to optimization have been independently proposed in the USA. In 1975, he was awarded (together with American scientist T. Koopmans) the Nobel Prize in economics «for the contribution into the theory of optimal use of resources».

The new, truly scientific approach to economics suggested by L. Kantorovich was incompatible with the official Soviet ideology. Economics was one of those humanities (along with history and philosophy) where the ideological conflicts were most critical and the struggle most tense.

From the side of the dogmatic economists, an open persecution of Leonid Kantorovich followed. During the whole his life, the Nobel Laureate Leonid Kantorovich was forced to struggle against the stupid and absurd opposition of orthodox Soviet economists, while persuading the only purpose – to help his country. The most part of his life Leonid Kantorovich devoted to the democratization of economics in his country, to the application of advanced science for the welfare of mankind.

Kantorovich coordinated a number of specific computing projects, including a nuclear studies project and a project on optimization of blanks nesting in the manufacturing industry. He realized the importance of computer technology at its very dawn and believed that this invention «will influence all human activities as greatly as did book-printing, steam engine, electricity, and radio». Fascinated with computers, he designed one of the first systems of automatic programming and a number of new computer configurations, some of which were then put into practice [2].

In 1957 Kantorovich was invited to the newly created Siberian Division of the USSR Academy of Sciences and was elected a Corresponding Member of the Academy in economics and statistics. Since then his main works were devoted to economical cybernetics.

#### 4. Search for Efficiency: Unconventional Siberian Architectures

Unfortunately, the poor Soviet technology, and the incompetence of the management left our country permanently behind the West in building and using computers.

In the beginning of '60s, it was clear to Siberian computer scientists that the only way to build high-performance system had to be in parallel computation.

Let us consider in brief some early Siberian architectural projects.

##### 4.1. Kantorovich's Contribution

The range of interests of Leonid Kantorovich was unusually broad, and he made original and deep contributions to all domains of his various activities.

Being an outstanding mathematician and author of fundamental mathematical theories, Kantorovich also paid considerable attention to the technical realization of complicated computational processes.

As early as the '40s, before the advent of electronic computers, Kantorovich successfully used tabulating machines for scientific calculations. He constructed at the Leningrad Branch of the Mathematical Institute of the USSR Academy of Sciences what would now be called a «multiprocessor parallel computing system» made of tabulators. With this system, the tabulation of Bessel functions by a special global program was carried out simultaneously for 121 functions.

Later, Kantorovich suggested enhancing the computational capabilities of tabulators by attaching what are now called «specialized processors». One such processor, designed for calculating functions, was created in the late '40s.

The concept of large-block organization of computational processes advanced by Kantorovich in the early '50s and developed later in numerous publications by him and his collaborators constituted his important contribution to computer science. Later on, some of the ideas of the large-block approach were developed in such programming languages as APL, PL-I, Algol-68, etc.

The large-block system with its operand files and dedicated bulk operations opens wide possibilities for new architectural solutions. These include, in particular, various forms of simultaneous processing of many elements or quantities. Thus, in organizing arithmetical operations on quantities, it is natural to use independent parallel processing of all components.

The idea of attached units «enhancing» computational possibilities of computers was formulated in its general scope in [3], where a computing system was proposed that consisted of a universal computer of conventional architecture and a small specialized computer designed for some definite kinds of bulk processing. It was assumed that the universal machine singled out large-block operators in the program and transferred them for execution to the attached unit. The high performance of the attached processor was obtained by exploiting peculiarities of various operators and operands exhibited by the architecture of the specialized small computer. To get the necessary adaptability and to enlarge the set of operations of the attached processor, microprogramming control, and a variable data format were used; for connection to the host computer a special interface unit was designed.

In the early '60s at the Institute of Mathematics of the Siberian Division where Kantorovich worked at that time, a project had been developed under his direction of an attached unit of this kind, which was called the «Arithmetic Mashine» (AM) and was intended primarily for speeding up the solution of problems in linear algebra and linear programming.

A pilot AM computer implemented in 1967–1968 was in operation at the Computing Center of the Siberian Division. For vector and matrix operations, its gain in speed, compared with universal computers of the same technological base, was one order. The AM computer seems to have been one of the first pipeline vector processors, and thus a prototype of modern supercomputers.

#### 4.2. *Homogeneous Computing Systems*

In 1962, in Novosibirsk, Edward Yevreinov suggested so called parallel Universal Computing Systems (UCS) with programmable structure [4]. The main features of UCS were:

- the basic element of UCS is a general purpose computer (Elementary Machine, EM);
- the UCS has a homogeneous structure, that is, it consists of identical, equally connected EMs;
- the number of EMs in the system can be changed;
- the instruction set, memory size and word length of an EM can also be changed.

It was proposed to distinguish the UCSs according to their topology as one-, two- and multi-dimensional; according to the type of exchange between EMs: parallel, sequential, and parallel-sequential; according to the spatial arrangement of EMs: concentrated and distributed.

In the Yevreinov's concept two levels of organization of parallel computing systems were considered: the macrostructural which has been briefly described above, and the microstructural concerning the inner structure of the elementary machines.

Here again a homogeneous approach was again proposed, based on so called Homogeneous Computing Media (HCM).

The main properties of the HCM are: homogeneity; short-range interaction; universality of the cells; possibility of setting each cell to implement any function from the chosen universal set.

According to Yevreinov, the HCM should be manufactured in a single technological process, like some «computing tissue», getting the required «pattern» at the last stage of production by means of appropriate configuring.

The main idea of homogeneous media was embedding of logical nets, that is, realization of arbitrary automata, into a planar homogeneous structure shaping the cells in a corresponding manner.

In some sense, the ideas of Yevreinov anticipated by far the present state of computer science and outlined most of the fundamental problems of development of high-performance computing systems.

#### 4.3. *Distributed Functional Structures*

Most of the cellular automata models (including Yevreinov's HCM) are universal. They can realize arbitrary functions and algorithms, and the synthesis of necessary logical structures proceeds using classical automata theory techniques. Unfortunately, most specific functions will incur time and hardware redundancy when implemented in this way.

Specialized homogeneous structures, which immediately map algorithms into circuits, represent an alternative to the universal ones. In these structures, the given algorithm is simulated by signal propagation through a specialized logical net. A classical example of such structure is the content-addressed, or associative memory with its special basic operation of «equality search». Other specialized structures realizing other basic operations have emerged as well.

In 1971, a specialized cellular array (called  $\alpha$ -structure) having the basic operation of «extreme search» was proposed in Novosibirsk. Later on, numerous arrays have been designed implementing various basic operations (threshold searches, nearest neighbor searches, compression, etc.). Arrays of this type have been called Distributed Functional Structures (DF-structures) [5].

An important feature of the DF-structures is their multifunctionality. Thus, an  $\alpha$ -structure can be efficiently used not only for extreme selection, but also as an associative memory, a programmable logic array, an interconnection network, etc.

The conception of DF-structures allows to design efficient parallel accelerators for diverse computer architectures. Indeed, the modern technology allows to implement distributed functional arrays of sufficient size, which can become a new type of VLSI product, Cellular Microprocessors.

## Conclusion

The Siberian Branch was conceived, from the very beginning, as a strong mathematical and cybernetical scientific center. During a certain period, the Novosibirsk Akademgorodok was a kind of a capital of Soviet Computer Science.

Turning back to the history of formation and development of computer science in Siberia, we can suppose that it was not surprisingly that just here, in the Siberian branch of the RAS, recently an initiative came in existence of exploring the history of computer science. During several years, we have published (in Novosibirsk, as well as in Moscow) a series of scientific volumes on the history of Russian Computer Science. Several topics from these books have been presented in this paper.

To-day, our investigations are in progress.

## References

1. Aleksey Andreevich Lyapunov / Edited by N. Lyapunova and Ya. Fet. Novosibirsk: Scientific Publ. Center of RAS, 2001. 524 p. (In Russian).
2. Leonid Vital'evich Kantorovich: a Man and a Scientist. Vol. 1 / Edited by V. Kantorovich, S. Kutateladze, and Ya. Fet. Novosibirsk: Scientific Publ. Center of RAS, 2002. 544 p. (In Russian).
3. Kantorovich L.V. and Fet Ya.I. Computing System Comprising a Universal digital computer and a small digital computer / USSR Inventor's Certificate No. 172567, 1963.
4. Yevreinov E.V. and Kosarev Yu.G. High-performance homogeneous universal computing systems. Novosibirsk: Nauka, 1966. (In Russian).
5. Fet Ya.I. Parallel processing in cellular arrays / Tounton, England: Research Studies Press, 1995. 196 p.

## РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ ОБРАБОТКИ ДАННЫХ

**В.П. Шириков**

Объединенный институт ядерных исследований

Дубна, РФ

*shirikov@jinr.ru*

Создание и использование распределенных систем для компьютеризованной регистрации, передачи и обработки информации в интересах научных исследований началось меньше чем 50 лет назад: по-видимому, первая такая система в Советском Союзе была реализована в 1961 году для передачи по радиоканалам информации с объектов в Атлантическом океане для приема и обработки первым советским универсальным полупроводниковым компьютером ДНЕПР в Институте кибернетики, в Киеве. Основные работы по созданию многомашинных комплексов для распределенной обработки задач и данных развернулись в 60–70-х годах прошлого века практически одновременно во всех крупных исследовательских организациях и у нас, и за рубежом. Верхом их развития мы считаем сейчас Grid-системы, т.е. комплексы, удовлетворяющие, по определению одного из идеологов Grid Яна Фостера [1], следующим трем критериям:

- 1) система объединяет распределенные вычислительные ресурсы, которые не являются объектами исключительно централизованного локального управления (применения) и могут быть предоставлены по определенным правилам и соглашениям для внешнего доступа и использования;
- 2) система реализована с применением стандартных открытых протоколов общего назначения и интерфейсов (типа, возможно, OGSA/WSRF- архитектуры);

- 3) система гарантирует своим пользователям нетривиальное качество обслуживания (QoS, Quality of Service) по времени реакции, пропускной способности задач, доступности, разнообразию предоставляемых ресурсов, обеспечению сохранности информации и прав доступа к ней, так что в целом производительность, выгодность использования, полезность, отдача всей системы больше, чем от простой суммы ее составляющих.

В идеале такая Grid-система позволяет рядовому зарегистрированному пользователю ввести свою задачу для пакетной или интерактивной обработки, не адресуясь в паспорте задачи (ее JDL-файле) через атрибут типа Requirements к конкретной машине (CE, компьютерному элементу-серверу в составе вычислительных ресурсов Grid), если в составе программного обеспечения функционирования системы (ее middleware) есть сервисы высокого уровня типа RB (Resource Broker) и достаточный набор CE, как это имеет место в реализованных широкомасштабных проектах типа EGEE (Enabling Grids for E-sciencE, см. <http://lcg.web.cern.ch/LCG/>, <http://egee-jra1.web.cern.ch>), хотя, скажем, в американском проекте OSG (Open Science Grid) этого предусмотрено не было. На самом деле многие идеи, реализованные в современных Grid-структурах, не являются (как будет видно ниже) абсолютно новыми для тех, кто строил комплексы распараллеливания алгоритмов и распределенной обработки данных более чем 30 лет назад, применяя доступную в то время компьютерную технику и средства межмашинной связи, поэтому можно считать, что Grid – это во многом реализация старых идей на новой технологической основе.

В таком физическом научном центре на территории России, как Объединенный институт ядерных исследований (ОИЯИ), основанный в 1956 году и имевший тесные научные связи с западными физическими центрами, особенно с Европейским центром ядерных исследований (CERN, Женева), даже при отсутствии прямых межкомпьютерных связей возникали проблемы, требовавшие общего подхода к решению прикладных задач и совместимого программного обеспечения на вычислительных серверах, что облегчило бы по крайней мере если не обмен программами и данными по линиям связи, то хотя бы их переносом на магнитных лентах. Что касается основного типа обрабатываемой информации, то в ОИЯИ, CERN и других центрах (в Бруклине, Дэрисбури, Беркли, Ливерморе) это прежде всего были экспериментальные данные с детекторов на ускорительных установках. В ОИЯИ, где в равной степени решались задачи физики низких и промежуточных энергий, возникала и необходимость регистрации и обработки спектрометрической информации в исследованиях с применением реакторной техники. Тем самым одной из главных решавшихся с начала 60-х годов задач стало техническое и программное обеспечение всей цепочки съема, предварительной и окончательной обработки информации прежде всего для разных областей экспериментальной физики, а также компьютерная поддержка исследований в области теоретической физики. Способом решения этой задачи и стало построение распределенных многоуровневых вычислительных комплексов с применением перспективных магистрально-модульных систем и стандартизованных программных средств для решения прикладных задач: трансляторов с языков высокого уровня, библиотек программ общего и специального назначения, технологии распараллеливания алгоритмов, средств доступа пользователей к разнотипным вычислительным ресурсам. Этапы решения этой задачи представлены, в частности, в [2, 3, 4]. Основные работы в ОИЯИ, указанные ниже, были инициированы и выполнялись (до июля 1989 года) под общим руководством члена-корреспондента АН СССР Н.Н. Говоруна (начальник математического отдела ВЦ ОИЯИ (1963–1966), заместитель директора (1966–1988) и директор Лаборатории вычислительной техники и автоматизации (1988–1989)).

Началом работ по реализации распределенных многомашинных комплексов в ОИЯИ можно считать организацию в 1962 году простейшей связанной структуры из двух машин М-20 и КИЕВ [2, 4], установлен-

ных в ВЦ ОИЯИ и способных принимать по кабелю и обрабатывать информацию с анализатора спектров в измерительном центре Лаборатории нейтронной физики (ЛНФ), а также данные на перфолентах с измерительных полуавтоматов, на которых шла предварительная обработка фотографий событий в детекторах на ускорителе Лаборатории высоких энергий (ЛВЭ) ОИЯИ. В 1965 году эта структура была модернизирована за счет замены машины КИЕВ, которая была недостаточно надежной, на МИНСК-2 и установки второй ЭВМ М-20: машины класса М-20 стали основными для обработки данных и счета задач разного типа (схема доступна по адресу <http://lit.jinr.ru/LCTA/lcta30/img/schema1.gif>)

С 1966 по 1974 год проходил первый этап создания трехуровневого комплекса для реализации общих принципов и режимов, характерных и для современных распределенных систем:

- обеспечение экспериментов на линии с ЭВМ (нижний уровень);
- дистанционная пакетная и интерактивная обработка задач;
- интерактивный режим при подключении пользовательских терминальных устройств;
- распараллеливание алгоритмов на разных машинах комплекса;
- организация и использование банков программ и данных.

Реальность реализации этих принципов и режимов определялась появлением и приобретением целого набора ЭВМ нового типа: «малых» ЭВМ для on-line съема информации с экспериментальных установок или для использования в качестве групповых пользовательских устройств (RIOS: Remote Input / Output Stations) при формировании пакетов задач, передаче их вычислительным серверам и получении результатов счета в режиме дистанционной пакетной обработки; «средних» ЭВМ для предварительной обработки информации от аппаратуры измерительных центров и также поддержки режима RIOS и «больших» ЭВМ для использования в качестве базовых вычислительных серверов. В этот новый набор вошли машины ТРА (производство Венгрии с DEC-подобной системой команд), СМ-3/4, М-6000 (аналог HP-2116B), оригинальные HP-2116B, БЭСМ-3М и БЭСМ-4, МИНСК-2/22, CDC-1604A, БЭСМ-6, БЭСМ-6 становилась основным вычислительным сервером во всех научных организациях СССР, однако уже на стадии подготовки к ее полному использованию потребовалось развитие ее системного программного обеспечения: так родилась многоязычная мониторинговая система «ДУБНА» с набором трансляторов с языков ФОРТРАН, АЛГОЛ, РЕФАЛ, ПАСКАЛЬ, ЛИСП, CDL и др., а также библиотекой программ общего назначения, совместимой с библиотекой CERN и расширявшейся рядом новых программ, разработанных в ОИЯИ (история «одевания» БЭСМ-6 хорошо описана в публикациях [2, 3, 4]). Нужно отметить, что из указанного выше списка трансляторов только три делались непосредственно в Дубне (CDL и два варианта для ФОРТРАНа), остальные – в сотрудничающих организациях, откуда брались дополнительно и некоторые интересовавшие пользователей ОИЯИ библиотеки программ.

Условия применения БЭСМ-6 в ОИЯИ сразу диктовали необходимость развития ее технических и программных возможностей подключения разнообразных устройств ввода и архивирования информации, а также создания средств связи с другими машинами, особенно территориально удаленными. Результатом доработок в стойке управления внешними устройствами (УВУ) и операционной системе стало появление обслуживаемых операционной системой БЭСМ-6 восьми быстрых линий связи (с пропускной способностью до 500Кбайт/сек и возможностью подключения до 4 абонентских ЭВМ на каждой линии: удаленной техники измерительно-вычислительных центров ОИЯИ, выносных станций ввода-вывода для пакетной обработки задач (с местным названием ФС: Форттранные Станции, функции которых стали выполнять и ТРА с собственными внешними устройствами типа телетайпов, читающих устройств с перфокарт, магнитофонов и принтеров, а также машины типа МИНСК-2/22, М-6000, БЭСМ-4). Поскольку на машинах измери-

тельно-вычислительных центров велась предварительная обработка экспериментальной информации с расчетом на окончательную обработку с привлечением БЭСМ-6, то пришлось реализовать своеобразное распараллеливание алгоритмов обработки: это потребовало создания в рамках операционной системы БЭСМ-6 специального экстракода обслуживания линий связи и служебной задачи связи, а в ее мониторной системе – средств пропуска задач (с программами на стандартных алгоритмических языках типа ФОРТРАН), нуждающихся во внешних обменах информацией с программами предварительной обработки на периферийных ЭВМ. Использование аппарата внешних прерываний в БЭСМ-6 и периферийных машинах (таких как БЭСМ-4, где также был сделан в ОИЯИ аналог ФОРТРАН-ориентированной мониторной системы) облегчало реализацию аналога современного MPI (Message Passing Interface) сервиса, а следовательно, и указанного выше распараллеливания алгоритмов. Сложившаяся к концу описываемого периода схема, когда в 1972 году партнером БЭСМ-6 на верхнем уровне комплекса стала и машина CDC-6200, представлена в работе [2] и доступна по адресу <http://lit.jir.ru/LCTA/lcta30/img/schema2.gif>.

ОИЯИ не был единственной научной организацией на территории России, создававшей многомашинные комплексы уже в 60-е годы: можно отметить хотя бы создание системы коллективного пользования АИСТ в ВЦ СО АН, комплексы в ИПМ АН СССР, ВЦ АН, ИТЭФ, ИТМиВТ, ИФВЭ в Протвино. В чем-то по своим функциональным возможностям и система ОИЯИ, и другие упомянутые комплексы имели схожие черты с западными реализациями того времени: например, системой FOCUS в CERN, где была и машина-концентратор (CDC-3200) при базовых ЭВМ CDC-6500/6600, обслуживавшая 4 линии связи с 16 периферийными ЭВМ и пропускной способностью до 12Mbit/sec на каждой линии, и станции типа RIOS, и концентратор терминалов. Однако западным центрам было легче, им не пришлось, в частности, самим реализовывать ту системную среду типа ФОРТРАН-ориентированной мониторной системы, о которой речь шла выше. Что касается типа вычислительных серверов для обработки данных и проведения сложных расчетов с большой точностью, то физики этих центров всегда предпочитали использовать 60-битную технику, поэтому для целей съема информации с аппаратуры им было достаточно и 32-битных машин IBM, а для окончательной обработки применялись ЭВМ фирмы CDC, недоступные нам до 1972 года по причинам эмбарго: для нас появление в 1968 году 48-битной БЭСМ-6 уже было хорошим выходом. Нужно сразу сказать, что современный Grid тоже начался с активного применения кластерных PC-структур на основе 32-разрядной технологии, но при первых же поставках 64-разрядных процессоров типа AMD и Itanium 2 началось соответствующее перевооружение научных центров. Так, уже в 2003 году в CERN в рамках проекта Openlab [5] для освоения кластерной техники нового типа был сформирован кластер из 32 двухпроцессорных (на базе Itanium 2) узлов и началась адаптация прикладных пакетов программ типа PROOF (Parallel ROOT Facility) как версии популярного у физиков пакета ROOT для анализа больших объемов данных; в России примерно в то же время Институт вычислительной математики РАН начал использование подобного кластера с шиной Myrinet 2000 для решения задач численного моделирования, ранее решавшихся в основном через доступ в дистанционном режиме к суперкомпьютеру в Межведомственном центре РАН.

Период 1975 – начала 80-х годов был связан в ОИЯИ с расширением уровня базовых ЭВМ за счет внедрения ЕС ЭВМ, совместимых с IBM-360/370, и развитием средств терминального доступа к третьему уровню комплекса (см. схему по адресу <http://lit.jinr.ru/LCTA/lcta30/img/schema3.gif>).

Это развитие включало в себя создание специальных программируемых контроллеров (мультиплексоров) для «больших» ЕС ЭВМ с применением микропроцессоров, а также создание терминального концентратора на базе ЕС-1010 и программной подсистемы ИНТЕРКОМ, совместимой с аналогичной подсистемой INTERCOM для ЭВМ серии CDC-6000 в ОИЯИ, что позволило иметь:

- единый унифицированный язык INTERCOM и программно-технические средства для общения с разнотипными базовыми ЭВМ через буферную машину ЕС-1010 как концентратор терминалов, оснащенный собственными ресурсами внешней памяти на дисках и магнитных лентах, а также специальной операционной системой, разработанной в Лаборатории вычислительной техники и автоматизации (ЛВТА) ОИЯИ. Доступ к машинам типа ЕС-1060/1061/1066 поддерживался через указанные выше контроллеры и подсистему ТЕРМ (как расширение операционной системы этих машин, также разработанное в ЛВТА и понимавшее язык INTERCOM);
- независимый (от базовых ЭВМ) сервис для пользователей терминалов в процессе ввода, редактирования и архивирования информации при работе над текстом задач;
- выполнение директив пользователя по направлению подготовленной задачи на ту или иную базовую ЭВМ, по выдаче информации о состоянии очередей задач на той иной машине, по временной приостановке или прекращении выполнения задачи, выводу OUTPUT-файла задачи на печать или экран терминала.

В принципе концентратор по существу был способен выполнять функции тех программных сервисов, которые в современных Grid-системах относят к понятиям UI (User Interface), NS (Network Server) и RB (Resource Broker), особенно если пользователь обходился применением подмножества языка ФОРТРАН, общего для базовых ЭВМ, поскольку концентратор мог опрашивать состояние очередей задач на обслуживаемых машинах самостоятельно или по запросу пользователей терминалов. Возможность выбора для задачи той или иной машины облегчалась и тем, что все они (БЭСМ-6, CDC-6200/6500, ЕС-1060/1061 и впоследствии ЕС-1066) оснащались совместимыми версиями библиотек программ общего назначения.

С 1983 года подошло время массового внедрения стандартных средств организации локальных сетей в научных центрах и их подключения к внешним каналам связи. Что касается технологии и структуры, то возможный выбор касался сетевых технологий Token Ring/Token Passing или «молодой» Ethernet. В ОИЯИ локальная общеинститутская сеть заработала в 1985 году. Это была JINET (Joint Institute NETwork) с маркерным (token) методом доступа к магистрали на основе коаксиального 75-омного кабеля протяженностью порядка 12 км, прошедшего по всем основным корпусам ОИЯИ. В разных точках подключались сетевые узлы-адаптеры, каждый узел обслуживал от 8 до 12 последовательных (по стандарту RS-232C) портов для подключения терминалов (в том числе класса PC) или портов мультиплексоров ЭВМ. Для этой цели изначально было закуплено около 30 адаптеров от швейцарской фирмы Furrer-Gloor, они имели встроенные блоки памяти и микропроцессоры Z-80; через некоторое время в ОИЯИ было налажено собственное производство аналогичных адаптеров с добавочными возможностями, они также были задействованы в сети JINET. Внутреннее программное обеспечение узлов было полностью разработано в ЛВТА ОИЯИ по двум причинам: встроенное фирменное было слишком дорого для нас и, кроме того, мы предвидели необходимость модификаций, расширения его возможностей, а следовательно, нуждались как минимум в обладании исходными текстами, лучше – собственными.

Созданное программное наполнение узлов поддерживало образование виртуальных каналов связи (коммуникационных сессий) между абонентами через общий кабель с пропускной способностью около 1Mbit/sec. Каждый абонент (терминальный пользователь или компьютер) мог давать различные команды своему узлу: вызвать локальный редактор вводимого текста, проверить состояние других узлов или их портов, настроить свой порт на удобную для него скорость приема информации, указать свой признак конца вводимых строк и т.п., если определенные по умолчанию параметры его не устраивали. Если подобных указаний не требовалось, то абонент просто давал команду на образование виртуального канала связи

по определенному адресу (т.е. указав номер узла и порта либо символическое имя машины), начинал использовать этот канал и по завершении работы давал команду о разрыве соединения (конце сессии). Такие абоненты, как компьютеры, могли использовать подобные виртуальные каналы для обмена файлами средствами стандартной программы KERMIT. Специальный узел-администратор и его партнер («горячий резерв») автоматически контролировали перемещения маркера (token`а), собирали статистику соединений узлов и ошибок, обнаруживали возможную пропажу маркера и обеспечивали его восстановление, могли выключить какой-то узел из обслуживания по причине замеченных отказов или ошибок в работе, делая запись об этих своих действиях в файле статистики для технического персонала, ответственного за работу сети.

В 1988 году JINET стала абонентом IASNET, международной сети коммутации пакетов по протоколу X.25, будучи подключенной наземным каналом связи к коммуникационному узлу в московском институте ВНИИПАС [4]. Это открыло возможность прямого обмена файлами с отечественными и зарубежными физическими центрами, использования режима telnet для доступа к тем или иным внешним вычислительным серверам (многие физики ОИЯИ имели пароли для такого доступа, будучи членами различных международных коллабораций). Через год в ОИЯИ уже заработала и первая очередь стандартной сети Ethernet, она была состыкована с сетью JINET, как это представлено в схеме, доступной по адресу <http://lit.jinr.ru/LCTA/lcta30/img/scema4.gif>. Это было время, когда БЭСМ-6 была снята с эксплуатации и новые серверы подключены к Ethernet: кластер из двух машин VAX-8350 с общей дисковой памятью (около 12 Gbyte), устройствами для картриджей (по 200 Mbyte), магнитофонами для лент с плотностью записи до 6250 bit/inch; достаточно мощная станция MEGATEK/WIZZARD 7555 для нуждающихся в наличии трехмерной графики (например, для физиков, просматривавших заинтересовавшие их после программной обработки сложные события (events) в детекторах на каналах вывода пучка из ускорителя).

В начале 90-х годов пропускная способность российских наземных каналов связи была неадекватной повышенным требованиям распределенного обмена и обработки информации для задач физики высоких энергий: практически она была в пределах 32Kbit/sec. По этой причине в ОИЯИ были организованы и начали эксплуатироваться в 1994 году два прямых спутниковых канала для доступа к сетям DFN (Германия) и HEPNET (High Energy Physics NETwork) с точкой входа в Италии (INFN, Итальянский институт ядерных исследований, в свою очередь имевший прямой наземный канал с CERN). Пропускная способность каждого из этих каналов была порядка 64Kbit/sec, мы использовали антенну и советский спутник РАДУГА для канала на Германию, а для итальянского канала – наземную связь с рядом расположенной станцией космической связи «Дубна» (CRC-2) и международный спутник системы INTELSAT. Мы планировали возможность предоставить и другим физическим центрам на территории России возможность воспользоваться вместе с ОИЯИ этим вторым каналом, как это представлено схемой в публикации [2] или по адресу <http://lit.jinr.ru/LCTA/lcta30/img/schema5.gif>. Конечно, во многих случаях наземные каналы связи предпочтительнее спутниковых с той же пропускной способностью, поэтому, когда предоставилась возможность увеличить их пропускную способность (сейчас она порядка 1Gbit/sec), мы воспользовались этой возможностью и прекратили использовать спутниковые каналы.

Наши ближайшие перспективы в использовании распределенных систем – в реализации проекта RDIG (Russian Data Intensive Grid) как части международного проекта WLCG (Worldwide LHC Computing Grid [6]); 8 институтов (в Москве, Московской области, Петербурге) стали основными участниками от Российской Федерации в этом проекте: ОИЯИ, ИФВЭ, ИПМ, РНЦ КИ (Российский научный центр «Курчатовский Институт»), НИИЯФ МГУ, ИТЭФ, ПИЯФ (Петербургский институт ядерной физики в Гатчине) и ИМПБ (Институт математических проблем в биологии).

### Благодарности

Искренняя благодарность людям (особенно Georg Trogeman, Alexander N. Nitussov, Wolfgang Ernst), организовавшим публикацию книги, упомянутой в ссылках [3, 4] и содержащей массу информации по компьютерингу и средствам связи в Советском Союзе, а также организациям, поддержавшим это издание: Ministry for Schools, Education, Science and Research of the German state North-Rhine Westfalia и Academy of Media Arts Cologne.

### Библиографический список

1. Foster I. What is the Grid? A Three Point Checklist: <http://www.gridtoday.com/02/0722/100136.html>
2. Шириков В.П. Математическое обеспечение вычислительных комплексов и сетей // Программирование. № 3. 1991. С. 15–28.
3. Apokin I.A. The Development of Electronic Computers in the USSR // Computing in Russia. The History of Computer Devices and Information Technology revealed. Vieweg, 2001.
4. Shirikov V.P. Scientific Computer Networks in the Soviet Union // Computing in Russia. The History of Computer Devices and Information Technology revealed. Vieweg, 2001. P. 168–176.
5. The CERN openlab: a novel testbed for the Grid // CERN COURIER. October 2003. P. 31–32.
6. Memorandum of Understanding for Collaboration in the Development and Exploitation of the Worldwide LHC Computing Grid: <http://lcg.web.cern.ch/LCG/C-RRB/MoU/WLCGMoU.pdf>



## **С.С. ЛАВРОВ И ИСТОРИЯ ПРОГРАММИРОВАНИЯ В СССР**

**И.Р. Агамирзян**

Майкрософт Россия и СНГ

Москва, РФ

*Igor.A@Microsoft.com*

Святослав Сергеевич Лавров был одним из основоположников программирования в СССР и руководителем группы, разработавшей первый промышленный транслятор Алгола 60. В докладе рассматривается научная деятельность С.С. Лаврова в контексте истории программирования в СССР.

## **DESIGNING OF THE COMPUTING SYSTEM FOR THE AUTOMATED CONTROL OF PRODUCTION PROCESSES**

**O.L. Ahremchik**

Tver State Technical University

Tver, Russia

*asremchik@mail.ru*

### **Introduction**

The fast updating of modern element base and software for construction of the automated control systems of production processes (ACS) requires a decrease of terms of their creation and intrusion. The temporary limitations and considerable dimension of a problem do not allow the designer to study in depth a field of application of means and systems of computer technology. That's why the necessity of creation of formal methods and techniques for an estimation of a possibility of standard engineering solutions on construction of computing systems, invariant to the type of a production process, is apparent. The consideration of historical stages of development of computing systems as a component part of ACS allows to dedicate their trial functions and ways of construction. The registration of the given information on incipient states of reduces number of iterations at a system model identification, its structural and parametric synthesizing, to speed up process of system creation and to decrease number of errors at decision making.

### **Structure and functions of the ACS**

ACS represents a multifunction outline. The hierarchy includes a set bound among themselves of systems: measuring, control, alarm and indication, computing, power supply, data transfer, actuating devices. The number component of ACS is determined by features of technology, machines and vehicles for implementation of one. Irrespective of number component each of them is under construction to multifunction tag and will realizes information, control and auxiliary functions. The information functions actuate: the assembly and processing of the current information about a course of a production process, calculation of technical and economic parameters, indication of the measured and counted values, alarm about their output for established limits, formation of the current and account documentation.

The control input and output information, archiving of the information, control of databases, system diagnostics are the auxiliary functions. The registration of a set of varied parameters on miscellaneous sites of a production process for calculation of control actions ensuring optimal control of process in whole is system-generative tag of ACS.

Presence of the computing system is a compulsory condition of ACS existence. It allows to supply the registration of communications between sites of a production process with the purpose of optimization of its object func-

tion. The computing system includes hardware, system and problem software, mathematical and algorithmic maintenance, orgware. Large dimension of the information vector are idiosyncrasy of the computing system.

### **History of computing system development as a component of the ACS**

The modern level of development of the architectural solutions is characterized by presence of a distributed network of devices of a lower layer of hierarchy, bound on the interface such as RS422/485. The devices are connected to serial ports by a computer, being an automated workstations of technologist, mechanics, power engineering personal, warehouse facilities personal. The computers are connected with each other on corporate or terrestrial network. The system has decentralized pattern.

On the initial stages of development (during 1970) the ACS creation started both finished by top hierarchical levels and was reduced to the installation of a computer of a general purpose. As a rule, a computer such as «CM» were used. The creation of communications with devices of measurement, control and actuating devices was integrated to considerable material costs. The system had the centralized pattern.

During 1980 the designing of control computer complexes implemented on the basis of a standard microcomputer (such as «ELEKTRONIKA 60», «DBK-2»), modules of linking with control object and human-machine interface. The complex was based on the general-system software and module structure of the problem software. The standard microcomputers appearance has not decided problems of linking computer with technology.

During 1990 the part of information and computing functions has passed to adjusting microprocessor controllers (such as «Remicont»). The controllers had the modules of linking with control object oriented on hooking up measuring and controllers, included in a structure of a state system of devices. They had the library of standard algorithms permitting to execute the configuration of the regulator in a structure of control systems. The communication between devices implemented with usage of specialized networks («Transit»). The given approach had mixed on-line – decentralized pattern. On a lower layer of hierarchy the decentralizing of functions on separate adjusting controllers was conducted. The personal computer PC XT/AT was used on top level of hierarchy.

The «Transit» networks were characterized by a small number of connected devices (up to 15) and necessity of applying of special airlocks for information exchange organization between regulators and personal computer. One implemented with usage of the interface «current loop». Length of communication circuits for data transfer from 10 to 500 meters under the speed of exchange 4800 bauds and high-noise immunities. Each controller in a network had the only system number.

At an intrusion in industrial production of computing systems on the basis of «Transit» have confronted with large difficulties. By the main reason was laying mining of channels interlayer communication on operational staff of firms which are not owning indispensable knowledge and skills. The large amount of information on algorithmic and software of local control instruments was required at creation of their configuration and installation under the computing system designing. The standard examples network and interlayer communication on the given period of development missed. In the order bookings practically there was not an information on connection of a computer and controller. Alongside with the main reason have had an effect absence of the problem software on a level of drivers for communication of regulators both computer and installation such as the interface in the unit of the airlock (RS232 or CL) with the help of strips (hardware).

The applying of control devices «Remicont» in a structure of the computing system would receive other development at presence of formal methods and CAD for computing system designing in a structure of ACS. But on that moment there was not generalizing properties of control instruments, communication equipment, supply unit, methods of a transmission of information on the series interface, program modules.

### Design stages of the computing system

The designing breaks down into the steps of the computing system creation as integrated system in view of interplay of its components (macrodesigning) and identification of functional subsystems of one (microdesigning).

The designing is reduced to selection of a computer, software (system and for activity with the equipment), system design of data transfer.

During the first design stage the structural and functional organization of the computing system is determined. The second one is contained the algorithmic and logical conditions of interplay component consideration. The third one is a selection of the parameters for an estimation of a system effectiveness are selected.

There is necessity to select hardware and program modules and the schemes for their electrical, logical and design implementation.

The purpose of the initial stages of designing consists in a component system definition, functional and territorial division between computing system components, organization of information, electrical and constructive compatibility between components.

The computing system is considered as distributed territorial hardware and software, bound communication channels for fulfillment of a data processing function. For maintenance of function ability on all territory the line of a power supply has to be made.

During decomposition of the computing system it is expedient to use for the registration of communications between separate subsystems principles of forecasting and coordination of interplays. The principle of prediction of interplay is encompass bayed organizations of control actions on subsystems of a lower layer in such a manner that each of subsystems becomes independent concerning other subsystems of the same level.

The principle of the coordination of interplays is bayed volume, that the siblings understand a binding signal as a padding variable of the solution.

### Description of the computing system on the initial stages of designing

The description of the computing system as whole is characterized by a structure of the system characteristics:

$$S = \langle C, F, S, P, H \rangle,$$

where: S – the computing system, C – communication of a system with an environment, F – functions, S – pattern, P – combination of properties, H – description of biotic cycle of a system during designing, operational testing, after retrofit, retargeting.

The models of each system characteristic are described for different levels of partition of hierarchic system organization.

The function – structural properties of members component a system, are described by expression:

$$\{ \langle F, S \rangle \}_r^m,$$

where: m – component number, r – hierarchical level number.

During designing the separation of a system from an environment implements on the basis of heuristic methods.

The function of a system is based on combination of functions of separate operations depicting intermediate transformations:

$$F \rightarrow \{f_1, f_2, f_3, \dots, f_i, \dots, f_l\}.$$

The function F is determined as transformation T of set of inputs  $\{X_i\}$  in set of outputs  $\{Y_j\}$  and is described by mapping:

$$F = T: \{X_i\} \rightarrow \{Y_j\}.$$

Each local function corresponds to the pattern block of a system. The implementation of the same function can be realized with applying of different structural organizations possessing the different technical and economic characteristics. This circumstance results in multi-variant approach of problems of synthesizing of the computing system on the basis of a given function.

The set of properties of a system characterizes its capacity to execute the functions, to save persistence of pattern, to pass in a new stable state at operating external disturbances. The technical implementation of these properties is connected to costs of material means expressed in categories of the cost, weight, volume etc.

At different levels of partition property of computing system are described by the set of parameters:

$$P^k = \{P_1, P_2, \dots, P_i, \dots, P_n\},$$

where  $P_i$  – parameter depicting definite property of a level of partition.

The combination of properties will derivate a characteristics of a computing system.

## **ОСНОВНЫЕ ВЕХИ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ В СТРАНЕ ЧЕРЕЗ ПРИЗМУ ИСТОРИИ КАЗАНСКОГО ПРОИЗВОДСТВЕННОГО ОБЪЕДИНЕНИЯ ПО ВЫПУСКУ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ (КПО ВС)**

**М.Ш. Бадрутдинова**

Музей истории КПО ВС

Казань, РФ

[ava@icl.kazan.ru](mailto:ava@icl.kazan.ru)

История создания КПО ВС восходит к 1954 году – году создания Казанского завода математических машин (КЗММ), в дальнейшем переименованном в Казанский завод ЭВМ (КЗ ЭВМ).

Решение о строительстве этого предприятия было принято в мае 1951 года Постановлением СМ СССР, а в августе 1952 года определен генеральный подрядчик – Гипромашприбор (г. Ленинград). 4 августа 1954 года был назначен директор завода К.Е. Минеев и утверждено штатное расписание на 6 человек.

Завод строился с «чистого листа», строительная площадка располагалась далеко за городом на месте городской свалки и совхозных полей.

В октябре 1957 года был построен первый одноэтажный производственный корпус, а в июле 1958 – второй, в котором в миниатюре поместили весь завод. В этом же году выпущена первая продукция – приборы инфранизких частот (ИНЧ): НФ, НГПК, ДПВ, КВ, ЭСВ и приборы выдачи режимов резания (ВППР). Приборы ИНЧ во многих модификациях выпускались до конца 1968 года.

Такой подход позволил провести подготовку инженерно-технических работников и производственного персонала к работе с электронными и электромеханическими устройствами, которые необходимо было внедрять в серийное производство при освоении ЭВМ.

Это был наиболее сложный период становления нового завода – завода высоких технологий, т.к. параллельно с освоением новых изделий приходилось вести огромное строительство и оснащение производства под будущее серийное производство средств ВТ, а также создавать социальную базу для работников завода.

Только за период с 1957 по 1965 год под руководством основателя завода К.Е. Минеева были построены девять промышленных корпусов, инженерный корпус, котельная, склады, а также три жилых дома, два

детских комбината. Всего за историю завода было построено около 100 000 м<sup>2</sup> промышленных площадей на трех промплощадках. Станочный парк составлял более 2 000 станков, в том числе более 200 импортных, а также много уникального оборудования.

КПО ВС прошло четыре этапа развития. Первый этап связан с производством ЭВМ I поколения М-20, разработанной в ИТМ и ВТ АН СССР под руководством главного конструктора С.А. Лебедева. Параллельно с М-20 освоена единственная в стране и получившая широкую известность в мире ЭВМ с трехзначной логикой «Сетунь», разработанная в МГУ, главный конструктор Н.П. Брусенцов.

Второй этап связан с промышленным производством ЭВМ II поколения М-220, разработанной в НИИ электронного машиностроения под руководством главного конструктора В.С. Антонова, и ЭВМ М-220А, М-220М, М-222, разработанными специалистами СКБ завода.

Третий этап определяется промышленным производством ЭВМ «Ряд 1» Единой Системы (ЭВМ III поколения). Первой ЭВМ этого ряда стала ЭВМ ЕС-1030, разработанная в Ереванском НИИ математических машин под руководством главного конструктора М.А. Семерджяна. Необходимо отметить особое значение этой модели для завода как базовой модели становления новых промышленных технологий производства ЭВМ. Однако по технико-экономическим характеристикам по ряду причин эту модель нельзя назвать удачной. В 1976 году завод приступил к производству ЭВМ ЕС-1033, разработанной заводским СКБ под руководством главного конструктора В.Ф. Гусева. Период производства ЕС-1033 стал периодом самых высоких экономических показателей за всю историю существования завода. Технические решения по ЭВМ ЕС-1033 защищены авторскими свидетельствами (их более 100) в СССР и странах СЭВ, а также патентами США, Англии, Франции и ФРГ.

Четвертый этап связан с промышленным производством моделей «Ряд 2» и «Ряд 3» Единой Системы ЭВМ. Для дальнейшего развития ЕС ЭВМ было выбрано направление, ориентированное на применение элементной базы на основе энергоемких ЭСЛ-технологий. На Казанском производственном объединении вычислительных систем на этом этапе выпускались ЭВМ ЕС-1045, ЕС-1045.01, ЕС-1046 и вычислительные комплексы на их основе, разработанные в Ереванском НИИ математических машин под руководством главного конструктора А.Т. Кучука, а также терминальная ЭВМ ЕС-1007 и процессор телеобработки данных ПТД-5, разработанные в СКБ завода.

На протяжении всех четырех этапов завод выпускал целый ряд спецтехники. Этим направлением работ с 1959 по 1993 год был произведен выпуск разнообразных устройств печати. При этом завод был единственным в стране производителем широкой алфавитно-цифровой печати: АЦПУ-128-2, АЦПУ-128-3, ЕС-7032, ЕС-7036, ЕС-7037, ЕС-7038.

ЭВМ и вычислительные комплексы на базе ЭВМ заводского изготовления эксплуатировались в ВЦ академий наук всех республик бывшего СССР, а также в Госплане, Госстандарте, Государственном комитете по науке и технике, в Министерстве обороны и многих передовых предприятиях науки и техники страны.

Всего за 40-летнюю историю было выпущено 7425 ЭВМ, из них 519 ЭВМ поставлено на экспорт (страны СЭВ, Индия и т.д.). Основные данные и параметры этих ЭВМ приведены в приложении к настоящему сообщению.

Состояние информационных технологий и перспективы их развития невозможно оценить без анализа достижений и ошибок прошлого.

Строительство в Казани завода математических машин (КЗММ) положило начало развитию информационных технологий в Республике Татарстан (РТ).

За годы существования завода в РТ были созданы:

- ГНИПИ ВТ – Государственный научно-исследовательский и проектный институт по внедрению вычислительной техники в народное хозяйство;
- КНИТИ ВТ – Казанский научно-исследовательский технологический институт вычислительной техники по разработке и внедрению технологий производства средств вычислительной техники;
- КНУЦ – Казанский научно-учебный центр «Алгоритм» по подготовке и обучению специалистов по обслуживанию ЭВМ;
- факультет технической кибернетики и информатики в КГТУ им. А.Н. Туполева (КАИ);
- факультет вычислительной математики и кибернетики в КГУ им. В.И. Ульянова-Ленина;
- филиал Института проблем информатики АН СССР (ИПИАН), в дальнейшем преобразованный в институт проблем информатики АН РТ.

Таким образом, в республике был создан уникальный комплекс по подготовке кадров, разработке, производству и внедрению в народное хозяйство широкого спектра программных и аппаратных средств ВТ на основе прогрессивных технологий.

Вследствие этого республика занимала ведущее место в области ВТ и информатики в СССР и странах СЭВ.

Казанский завод ЭВМ и Казанское производственное объединение вычислительных систем на протяжении почти четверти века (1965–1990) были флагманом отечественного электронного машиностроения. Коллектив объединения, пройдя за сравнительно короткое время большой путь по созданию, освоению и производству ЭВМ, внес существенный вклад в ускорение научно-технического прогресса в стране, в укрепление ее обороноспособности.

В начале 90-х годов КПО ВС отличалось высокой культурой производства, наличием высокопрофессиональных инженерно-технических кадров с большим разработческим и внедренческим опытом, грамотными рабочими кадрами, первоклассным современным оборудованием, включая развитую исследовательскую и испытательную базы.

17 октября 1994 года решением руководства РТ КПО ВС было введено в состав нового объединения СВТ «Терминал». С этого момента завершилась история КПО ВС, которая является типичным позитивным примером создания и развития, а также типичным негативным примером уничтожения одного из флагманов информатики в стране.

В 1992 году КПО ВС завершило производство моделей «Ряд 2» и «Ряд 3» Единой Системы по ряду причин:

1. Субъективная ситуация в стране, определяемая включением России в общее информационное пространство, что требовало других технологических возможностей информационных отношений;
2. Слабость тогдашнего руководства предприятия, не сумевшего переориентировать предприятие в условиях передела собственности в стране;
3. Отсутствие технической политики со стороны вышестоящих организаций в новых экономических условиях.

С целью выхода из экономических трудностей периода перестройки в 1991 году на КПО ВС было создано совместное с Британской компанией ICL предприятие – ОАО «ICL – КПО ВС», в которое вошли лучшие инженерные кадры КПО ВС во главе с В.В. Дьячковым, который прошел путь от молодого специалиста до умелого руководителя, работая в наладочном производстве на различных должностях, начальником цеха КПО ВС, с 1998 года В.В. Дьячков возглавляет ОАО «ICL – КПО ВС» в должности генерального директора.

Накопленный опыт позволил на протяжении 15 лет создать творческий коллектив разработчиков в области информационных технологий с устойчивым рынком поставок информационных систем для МЧС РФ, МО РФ, Газпрома, министерств образования и внутренних дел, банковских систем и т.д.

И поэтому приятно сознавать, что огромный вклад КПО ВС в развитие отечественной вычислительной техники не пропал даром, а живет и развивается благодаря творческим усилиям воспитанников КПО ВС, ныне работающих в ОАО «ICL – КПО ВС», отличающегося стабильной экономикой.

**Перечень ЭВМ, выпускавшихся Казанским производственным объединением по выпуску средств вычислительной техники (КЗММ, КЗЭВМ, КПО ВС)**

| № | Наименование изделия                             | Шифр     | Период выпуска (г.) |       | Кол-во (шт.) | Технические характеристики                                                                                                                                                                                                                                                                                              |
|---|--------------------------------------------------|----------|---------------------|-------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   |                                                  |          | начало              | конец |              |                                                                                                                                                                                                                                                                                                                         |
| 1 | Универсальная электронная вычислительная машина  | M-20     | 1960                | 1964  | 63           | Производительность, опер./с: 18 тыс.<br>Емкость оперативной памяти, Мбайт: 4096<br>Цикл обращения, мкс: 4<br>Система команд: трехадресная<br>Структура слова, двоичных разрядов: 45<br>Число команд: 64<br>Емкость МБ: 32 Кслов<br>Емкость МЛ: 128 Кслов                                                                |
| 2 | Малая электронная цифровая вычислительная машина | «Сетунь» | 1961                | 1965  | 47           | Производительность, опер./с: 1 – 4,5 тыс.<br>Емкость оперативной памяти: 162 слова<br>Числа и команды представлены симметричным троичным кодом (1,0,-1)<br>Один разряд троичного кода: трит<br>Система команд: одноадресная<br>Структура слова: 9 тритов и 18 тритов<br>Количество команд: 24<br>Емкость МБ: 1944 слова |
| 3 | Малая электронная цифровая вычислительная машина | «Наири»  | 1965                | 1970  | 509          | Производительность, опер./с: 1,5 – 2 тыс.<br>Принцип управления: микропрограммный<br>Емкость оперативной памяти: 1024 слова<br>Структура слова, двоичных разрядов: 36<br>Система команд: двухадресная<br>Средства ввода и вывода: телеграфные аппараты РТА 50-2М и СТА-2М<br>Скорость в/вывода, зн./сек: 6–8            |
| 4 | Универсальная электронная вычислительная машина  | M-220    | 1965                | 1977  | 258          | Производительность, опер./с: 27 тыс.<br>Емкость МОЗУ на ферритовых сердечниках: 8192÷16384 машинных слов<br>Система команд: двоичная (42 двоичных разряда)<br>Емкость НМЛ (от 1 до 4 накопителей): от 4 до 16 млн. слов<br>Емкость НМБ: 24576 слов                                                                      |
| 5 | Универсальная электронная вычислительная машина  | M-222    | 1969                | 1978  | 551          | Производительность, опер./с: 27 тыс.<br>Емкость оперативной памяти: 32 Кслов<br>Цикл обращения: 6 мксек<br>Внешняя память:<br>Накопитель на МЛ,<br>млн. машинных слов: от 8 до 40<br>Емкость НМБ,<br>тыс. машинных слов: от 48 до 240                                                                                   |
| 6 | Универсальная электронная вычислительная машина  | ЕС-1030  | 1972                | 1976  | 286          | Производительность, опер./с: 70 тыс.<br>Емкость опер. памяти: 256÷512 Кбайт<br>Принцип управления: микропрограммный<br>Количество каналов: 1 мультиплексный, 3 селекторных                                                                                                                                              |

| №  | Наименование изделия                            | Шифр                                 | Период выпуска (г.) |       | Кол-во (шт.) | Технические характеристики                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----|-------------------------------------------------|--------------------------------------|---------------------|-------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                                                 |                                      | начало              | конец |              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 7  | Универсальная электронная вычислительная машина | ЕС-1033                              | 1976                | 1983  | 2300         | Производительность, опер./с: 200 тыс.<br>Емкость опер. памяти: 2048 Кбайт<br>Количество команд: 44 с ЕС ЭВМ «Ряд 1»<br>Количество каналов: 1 мультиплексный, 3 селекторных<br>Принцип управления: микропрограммный<br>Внешняя память: накопители на магн. дисках, накопители на магн. лентах                                                                                                                                                                                                                                                                                 |
| 8  | Универсальная электронная вычислительная машина | ЕС 3207<br>ЕС-1045<br>ОЗУ<br>ЕС-3203 | 1979                | 1981  | 50           | Производительность, опер./с:<br>для научно-технических задач – 870 тыс.<br>для экономических задач – 530 тыс.<br>Емкость оперативной памяти: 1 – 4 Мбайт (ферритовое ОЗУ)<br>Количество команд: 183 (ЕС ЭВМ «Ряд 2»)<br>Количество каналов: 1 – 2 байт-мультиплексных, 4 – 5 блок-мультиплексных                                                                                                                                                                                                                                                                             |
| 9  | Универсальная электронная вычислительная машина | ЕС-1045.01                           | 1981                | 1984  | 1716         | Производительность, опер./с:<br>для научно-технических задач – 860 тыс.<br>для экономических задач – 530 тыс.<br>Емкость оперативной памяти: 1 – 4 Мбайт<br>Количество команд: 183 ЕС ЭВМ «Ряд 2»<br>Количество каналов: 2 байт-мультиплексных, 4 блок-мультиплексных                                                                                                                                                                                                                                                                                                        |
| 10 | Универсальная электронная вычислительная машина | ЕС-1046                              | 1984                | 1992  | 1627         | Производительность, опер./с:<br>для научно-технических задач – 1300 тыс.<br>для экономических задач – 660 тыс.<br>Емкость оперативной памяти: 4 – 8 Мбайт<br>Система команд: универсальная ЕС ЭВМ «Ряд 3»<br>Количество каналов: 2 байт-мультиплексных со скоростью передачи в монопольном режиме 160 кбайт/с; 4 блок-мультиплексных со скоростью передачи одного канала до 3 Мбайт/с<br>Принцип управления: аппаратно-микропрограммный<br>Емкость упр. памяти: 64 кбайт<br>Пульт управления: вместо механического применен сервисный процессор<br>Система самоконтроля: 95% |
| 11 | Универсальная электронная вычислительная машина | «Урал-11Б»                           | 1966                | 1966  | 8            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 12 | Терминальная электронная вычислительная машина  | ЕС-1007                              | 1985                | 1990  | 251          | Производительность, опер./с:<br>для научно-технических задач – 100 тыс.<br>для экономических задач – 50 тыс.<br>Емкость оперативной памяти: 1 Мбайт<br>Универсальный набор команд: ЕС ЭВМ «Ряд 3»<br>Микропрограммная поддержка системы виртуальных машин, пульт оператора включает сервисный процессор и адаптер сетевой сервисный                                                                                                                                                                                                                                          |

**Перечень вычислительных и управляющих комплексов на базе ЭВМ, выпускавшихся Казанским производственным объединением по выпуску средств вычислительной техники (КЗММ, КЗЭВМ, КПО ВС)**

| № | Наименование изделия                     | Шифр    | Период выпуска (г. г.) |       | Примечание      |
|---|------------------------------------------|---------|------------------------|-------|-----------------|
|   |                                          |         | начало                 | конец |                 |
| 1 | Вычислительный комплекс                  | ВК-1010 | 1974                   | 1977  | На базе ЕС-1030 |
| 2 | Управляющий комплекс                     | УК-1010 | 1976                   | 1990  | На базе ЕС-1030 |
| 3 | Вычислительный комплекс                  | ВК-1033 | 1978                   | 1985  | На базе ЕС-1033 |
| 4 | Вычислительный комплекс двухмашинный     | ВК-2М45 | 1982                   | 1987  | На базе ЕС-1045 |
| 5 | Вычислительный комплекс двухпроцессорный | ВК-2П45 | 1982                   | 1986  | На базе ЕС-1045 |
| 6 | Вычислительный комплекс двухмашинный     | ВК-2М46 | 1986                   | 1992  | На базе ЕС-1046 |

## РАЗВИТИЕ СЕТИ ПЕРЕДАЧИ ДАННЫХ СО РАН: ИТОГИ ТРЕХ ДЕСЯТИЛЕТИЙ

**С.Д. Белов, Л.Б. Чубаров, А.М. Федотов, В.С. Никульцев**

Институт ядерной физики им. Будкера СО РАН

Новосибирск, РФ

[belov@inp.nsk.su](mailto:belov@inp.nsk.su)

Организации удаленного доступа к вычислительным ресурсам и распределенной обработке данных уделялось значительное внимание на протяжении всей истории развития вычислительных технологий в СО АН СССР, а затем в СО РАН. Проекты середины 70-х годов «Академсеть», ВЦ КП заложили основы сетевой инфраструктуры СО РАН и ННЦ. Эта инфраструктура была использована в первой половине 90-х для построения IP-сети с использованием современного телекоммуникационного оборудования. В начале 2000-х была проведена значительная модернизация сети. В настоящее время сеть передачи данных СО РАН объединяет научные институты и организации СО РАН, СО РАМН, СО РАСХН, ГНЦ вирусологии и биотехнологии «Вектор», а также ряд других научных, учебных, медицинских организаций, учреждений культуры и социальной сферы. СПД СО РАН является одним из важнейших факторов, определяющих эффективность работы всего Отделения. Предоставляя услуги более 150 организациям-абонентам, обслуживая более 40000 пользователей только в ННЦ и порядка 30000 в региональных научных центрах, объединяющая около 15000 компьютеров, СПД СО РАН стала крупнейшей корпоративной научно-образовательной сетью России. В докладе приводится информация о технических характеристиках сети, о правилах доступа к ее информационным и вычислительным ресурсам, о принципах, гарантирующих безопасность. Рассмотрены перспективы и направления развития СПД СО РАН.

## ЭЛЕКТРОННАЯ ПОДГОТОВКА ИЗДАНИЙ

**А.А. Берс**

Институт систем информатики им. А.П. Ершова СО РАН

Новосибирск, РФ

[baebers@iis.nsk.su](mailto:baebers@iis.nsk.su)

Все началось с того, что для издания «Пересмотренного сообщения об АЛГОЛе-68» ИФИП предоставила готовые фотоформы оригинала описания языка, что позволяло *безошибочно* воспроизвести английскую

часть, но потребовало применить те же шрифты (шесть различных начертаний для трех гарнитур) и в русской части.

Работа по подготовке качественного издания билингвы на английском и русском языках с по-абзацным соответствием потребовала очень много времени и оказалась высшей категории сложности. Перед нами предстал воочию *живой пример* трудоемкого *ручного* использования «системы *автоматизированного* фотонабора на ЭВМ».

Прежде чем двигаться далее, следует сориентироваться в общем контексте состояния дел с ЭВМ того времени. В стране начался серийный выпуск ЕС ЭВМ, главным образом ее младших моделей ЕС-1020 и ЕС-1030, а мини-ЭВМ серии СМ только планировались. Практически везде машины использовались в режиме пакетной обработки задач, о системах коллективного доступа и диалога с машиной еще только разговаривали, дисплеи воочию видели единицы. Устройствами ввода-вывода были перфокарты и перфоленты, печать через телетайпы и АЦПУ с весьма ограниченным набором символов, оперативная память была маленькой (32–128 Кбайт), сменные магнитные диски – 7,5 Мбайт, вся внешняя архивная память на магнитных лентах. Стоит вспомнить, что микропроцессор Intel-8080 был построен в 1974 году, первый персональный компьютер – в 1976, а IBM PC XT был выпущен в 1981 году.

На основе полученного опыта в 1976 году в ВЦ СО АН СССР была выдвинута концепция «Электронной подготовки изданий», которая реализовывалась в двух проектах: САПФИР (Система автоматизированной подготовки фотонаборных изданий с редактированием) – (1976–1979) для Первой образцовой типографии в Москве; РУБИН (Редактирование, управление, база информации, набор) – (1979–1989) для Издательства газеты «Правда».

По адресу <http://ershov.iis.nsk.su/russian/> доступны полные тексты материалов, хранящиеся в архиве А.П. Ершова, в разделе Программные проекты: САПФИР, РУБИН, АСТРА, а их печатное подробное изложение опубликовано в сборнике «Новосибирская школа программирования. Переключки времен». (Новосибирск, 2004).

Сегодня (на время доклада) эти материалы с очевидностью демонстрируют желание дать издательским редакторам возможность работать по-человечески даже на тех аппаратных и технических средствах и пакетных режимах, которые были в наличии.

Конечно, дальше опытной эксплуатации САПФИРа (пакетной системы редактирования на ЕС-1020 в операционной системе ДОС) дело не пошло, как и для ее терминальной версии для ОС ЕС ЭВМ – системы АСТРА.

Естественно, что работы начинались с системного анализа положения дел в предметной области – полиграфии. Для САПФИРа это был «Эскизный проект», а для РУБИНа – «Системный анализ производственных процессов по выпуску газеты «Правда». Этот анализ охватывал не только типы и структуру объектов и процедур их обработки, но и систему взаимосвязей различных видов деятельности в отрасли.

При этом в качестве исходных приняты два следующих взаимно дополнительных тезиса:

- Всякая *деятельность* является нормированной (Щедровицкий, [7, с. 156–196]), т.е. *проводится по программе*;
- Всякое *единичное исполнение* программного фрагмента в соответствующей операционной обстановке компьютером *является* (Берс, [1]) не только моделью деятельности, но и самой настоящей *конкретной деятельностью*.

Появление компьютеров как нового класса субъектов, способных осуществлять весьма сложные типы деятельности, создало предпосылки для конструктивного моделирования и изучения циклов жизнедеятельности.

тельности и поведения субъектов. В частности, сегодня кроме людей только компьютеры в состоянии реализовать овеществленную в знаках в форме программ и других текстов деятельность.

Известны два способа создать много одинаковых вещей:

1. Создадим первую требуемую вещь (оригинал), а затем проведем эту деятельность еще раз, а если нужно, – то еще... и еще... В этом случае характеристики копии почти соответствуют характеристикам оригинала – будем называть этот способ *повторением*.
2. Вместо того чтобы делать нужную вещь, создадим нечто дополнительное (противоположное ей) – оригинал, а затем будем повторно производить некоторую другую и более простую деятельность, которую назовем *тиражированием*.

Заметим, что при тиражировании копии одинаковы, а многократное повторение можно обеспечить, сделав вместо требуемой вещи нечто другое – машину.

Великое изобретение Иоганна Гуттенберга – практически полезный способ книгопечатания – по сути своей состояло в том, что он трижды применил к процессу издания тиражирование с тем, чтобы все соответствующие тексты были бы одинаковыми:

- единожды гравировался пуансон вместо многократного нарезания одинаковых матриц;
- по одной матрице отливается много одинаковых литер (зеркальных к буквам);
- составляется одна рельефная печатная форма – дополнительная к печатаемой полосе текста.

Гуттенберг позаботился, чтобы в третий оригинал (печатную форму) можно было сравнительно легко вносить исправления, однако при дальнейшем следовании его способу с целью увеличения размеров тиража – матрицировании и отливке стереотипов – уже получается неизменяемый оригинал.

Только в настоящее время, после развития компьютеров, смог возникнуть гипертекст, что позволило реализовать «живой» текст. Гипертекст позволяет объединить обычный языковой текст и другие формы представлений (изображение, звук, анимацию). Гипертекст обеспечивает разделение понятий копирования в реальном мире, где копия всегда теряет связь с оригиналом, и копирования в знаковом мире, в котором представление все время воспроизводит оригинал.

С другой стороны, пользуясь современными гипертекстовыми средствами великого Интернета, вполне легко выставить на сайте текстовый фрагмент, доступный из любого места Сети именно посредством повторения деятельности по его показу, вследствие чего любое изменение данного фрагмента будет отслеживаться при каждом обращении к нему. Полное описание такой деятельности (доступ по Сети плюс так называемый рендеринг текста) будет весьма сложным, а ее безошибочное осуществление человеком и во все невообразимо. Тем не менее она действительно буквально повторяется каждый раз, когда кто-то где-то видит данный фрагмент текста, который может фактически находиться за тысячи километров.

Заметим, что при обсуждении семантики деятельности почти всегда упускают из виду, что все языковые конструкции, в том числе тексты изданий и программы, суть системы знаков и находятся в особом знаковом мире. Однако еще в 1960 году в работах Московского методологического кружка [7] было показано, что свойства и связи вещей (т.е. отдельных целостностей реального мира) не только не совпадают со свойствами и связями их образов в знаковом мире – объектов, но и не являются их параллельным (изоморфным) отображением.

В ходе единичного исполнения некоторой конкретной деятельности субъектом можно обеспечить запись о выполняемой последовательности предписаний программы, включая доступы к используемым при этом объектам – «ход» исполнения, и ее размещение в знаковом мире. Некоторый другой субъект может воспринять эту запись как программу для своей деятельности и, возможно, сможет ее исполнить.

В таком контексте запись можно трактовать как текст, а исполнение соответствующей деятельности считать смыслом, который второму из субъектов удалось извлечь из этого текста. Другими словами, смысл (вос)создается субъектом – читателем. Наш язык отражает эту типовую ситуацию присказкой: «Каждый понимает в меру своей...»

Поскольку Субъект должен содержать в себе хотя бы один активный элемент, память и, по крайней мере, некоторые из операционных обстановок, то субъект не может быть вложен в знаковый мир [1]. Скорее наоборот, именно субъект «смотрит» из реального мира в знаковый мир. Осуществляя связи объектов знакового мира и вещей реального мира и выделяя конкретный знаковый подмир, субъект делает это, используя подходящие формы измерительной деятельности как доступы в реальный мир.

При этом в знаковом мире могут находиться вызовы, программы которых лежат во внутреннем мире субъекта, и тем самым в знаковом мире выглядят как элементарные. Точно так же во внутреннем мире субъекта могут находиться знаки, относящиеся к сущностям, лежащим вне него. Поскольку единичное исполнение всегда происходит в некоторой определенной операционной обстановке, то саму программу можно трактовать и как текст и как свертку порождаемых ею текстов-ходов. Соответственно можно говорить о множестве смыслов, связанных с программой-текстом.

Другими словами, программы – это новый, более мощный и богатый класс текстов, что оправдывает «интервенцию» программистов в полиграфию. Полагаем, что текст есть линейная упорядоченная последовательность символов, принадлежащих объединению видимого и невидимого алфавитов, а собственно текст является видимым.

Сегодня в гипертекстах это называется тегами (разметкой), причем невидимыми становятся все символы, заключенные в скобки невидимости, для которых используются «<» и «>».

Характеристическими чертами книги как объекта, которые должны быть сохранены в системы подготовки изданий с помощью ЭВМ, являются, по нашему мнению, следующие:

- Обязательное соответствие и верность оригиналу – т.е. инвариантность собственно текста при переизданиях, что, по-видимому, не требует разъяснений.
- Однородность исполнения одинаковых элементов издания – это требование проходит «красной нитью» через все руководящие материалы и методические пособия по редактированию и полиграфическому оформлению книжных изданий, те же требования однородности применяются для многотомных или серийных изданий.
- Качественность издания – это относится к каждой книге, ибо она не только продукт производства, но и произведение искусства. Выбор формата и полей полосы, шрифты, иллюстрации и подписи к ним, способ оформления примечаний, уравновешенность композиции спусков, организация указателей, оглавления, выходных данных и, конечно же, титульного разворота – все это предмет заботы не только художника, художественного и технического редакторов, но и всех других людей, выпускающих книгу.

Особо важным представлялось обеспечение комплексного характера полной работы над изданием для всех этапов работ с помощью ЭВМ и накопление в архиве системы не только подготовленных собственно текстов, но и отдельно запаасаемых сформированных способов полиграфического исполнения изданий, о чем ранее и не мечталось.

Система «Электронной подготовки изданий» должна обеспечить реализацию указанных выше характеристик издания, удобство пользования и эффективность работы и обладать следующими свойствами:

- Подготовка издания должна начинаться с ввода авторского текста в память ЭВМ и завершаться выдачей комплекта форм всех полос книги. Только так можно обеспечить эффективное использование систем и надежность процессов переработки текста.
- Обработка издания в ЭВМ должна включать в себя осуществление всей правки, как представленной издательством, так и вызванной неизбежными ошибками операторов-наборщиков, а также реализацию всех указаний технического и художественного редактора, относящихся и к особенностям текста, например выделения, набор формул и т.п., и к его расположению на полосах, например размещение иллюстраций, организация заголовков и т.п.
- Легкость внесения изменений в текст и его оформление в течение всего процесса подготовки издания на ЭВМ. Рассматриваемая вместе со способностью программ возможность безошибочной реализации подстановок в текст даже для достаточно сложных контекстных условий, управляющих такой подстановкой, что позволяет накапливать улучшения в тексте издания в процессе его подготовки в ЭВМ. Вместе с комплексностью и возможностью обеспечить защиту частей текста от непреднамеренного вмешательства и повреждения, которую предоставляют современные системы программирования, это приведет к качественно новому уровню работы.
- Необходимо осуществить отделение текста издания от его полиграфического оформления, благодаря чему появится возможность работать по-новому в сравнении с существующей технологией.
- Накопление в системе общих приемов полиграфического оформления и средств осуществления элементов изданий и возможность использовать их простым указанием на их названия. Эти приемы и правила, накопленные за века развития книгоиздания, разбросаны по различным методическим и учебным пособиям по полиграфии или же передаются «из уст в уста» среди полиграфических и издательских работников. Собрание этих приемов и системная организация их применения позволяет после включения их в библиотеку системы сделать применение весьма сложных способов организации текста доступным даже рядовому пользователю.
- Правильное разделение работ, осуществляемых системой, между пользователем-человеком и ЭВМ, с учетом того, что программы для ЭВМ не могут в ходе своей работы ни опираться на смысл обрабатываемого текста, ни воспринимать художественные образы, задуманные оформителем или художественным редактором. Все решения, принимаемые на основе такой информации, должны делаться человеком, сообщаящим затем машине, что нужно сделать для реализации этих решений, поскольку ЭВМ лучше человека приспособлена для выполнения однообразной рутинной технической работы. Правильный учет указанных факторов позволяет организовать выполнение работ в два этапа – однократное, разовое, принятие решения человеком и систематическое, монотонное осуществление принятого решения с помощью ЭВМ.
- Отсюда логически следует, что, в принципе, все процессы, связанные с набором и подготовкой оригинала печатных форм, могут быть вынесены за пределы типографий. Другими словами, при применении компьютеров для издания чем раньше текст будет введен в память ЭВМ, тем эффективнее и надежнее будут проводиться процессы его обработки и тем быстрее и с меньшими усилиями будет получен искомый результат.
- Однако следует заметить, что реализация этого принципа потребует значительной реорганизации существующих производственных отношений, что, конечно, значительно сложнее, чем построить автоматизированную систему подготовки фотонаборных изданий.

(Вышеприведенный перечень – это цитата из «Эскизного проекта системы САПФИР» 1977 года.)

Перечислим те свойства системы САПФИР, для которых не было найдено аналогов в доступной литературе того времени.

1. Последовательное проведение разделения собственно текстовой и полиграфической информации.
2. Комплексный подход к подготовке изданий, опирающийся на ввод в первую очередь исходного текста издания для хранения в архиве системы и накопления в нем всех последующих результатов подготовки.
3. Подход к верстке текста с предварительным расчетом макета издания.
4. Овеществление и накопление в системе общих приемов полиграфического оформления структуры и элементов изданий и возможность их использовать в последующем.

Преимущества, которые дает разделение текста и его полиграфического исполнения в системе автоматизированной подготовки изданий, проявляются на всех уровнях процесса подготовки издания: на редакторском уровне – отсутствие опасности при смене оформления испортить текст. На издательском уровне – возникает возможность легкого, без повторения всех работ по набору и выверке текста, переиздания книги в другом полиграфическом варианте; на уровне оператора-наборщика – поскольку связь элементов, составляющих текст с их полиграфическим оформлением, устанавливается однажды заранее, то объем набираемой информации значительно уменьшается.

Простота и экономичность получения вариантов освобождает редактора (а также технического и художественного редакторов) от «бремени уже сделанной работы», позволяя им ставить эксперименты для выбора наилучшего способа оформления текста, проверки их путем выдачи пробных страниц через фотонаборный автомат и возможности сравнить варианты перед принятием окончательного решения. В общем случае – чем сложнее текст и оформление подготавливаемого издания, тем больше выгоды от применения автоматизированной системы для его обработки. Справедливость этих положений тем больше, чем сложнее набираемый текст. Особенно большая экономия получается при наборе формул и таблиц, т.е. блочных текстов.

В общем случае строение команды текстового редактирования имело следующий вид:

в [диапазоне] при [условии] выполнить [действие],

где *диапазон* указывается парой граничных адресов, условие может быть задано образцом, а *действие* выбирается из списка команд текстового редактирования.

К средствам редактирования относится также язык описания блочных (двумерных) текстов.

В состав средств редактирования были включены форматные директивы (форды), которые могут быть вставлены в текст соответствующими командами. С помощью фордов осуществляется вызов процедур управления расположением и форматом текста, находящихся в системной библиотеке фордов (теперь такие вещи называются стилями). Вызов этих форматных процедур мог производиться также неявно через связи, устанавливаемые в спецификации заказа между некоторым типом элементов структуры текста и соответствующей процедурой.

Верстка производится по макету. Для каждой полосы рассчитываемого макета собиралась глобальная информация о тексте. Процессор верстки-выключки организован таким образом, чтобы локализовать на компенсационной основе отклонения от ранее рассчитанного макета в пределах минимально возможного числа абзацев. По завершении расчетов верстки происходит окончательное формирование полос.

### **Система РУБИН газеты «Правда»**

При защите Эскизного проекта системы САПФИР на техническом совете Госкомиздата СССР А.П. Ершова взял за жилетную пуговицу директор издательства ЦК КПСС «Правда» Борис Александрович Фельдман и сказал: «Я хочу, чтобы у нас тоже была такая система. И даже – лучше»; так началась история проекта РУБИН.

Прежде всего предусматривалось проведение системного анализа производственной деятельности комбината с позиций информатики и вычислительного дела, и 6 июня 1977 года на территорию комбината «Правда» для детального изучения производственных процессов по выработанной методике высадили «десант СО АН СССР» в составе: А.А. Берс, Г.Д. Чинин, В.В. Грушецкий, Ю.А. Первин, П.К. Леонов, Ф.Р. Цанг.

Разработчики довольно быстро осознали, что комбинат в целом – это огромное всесоюзное объединение, издающее и печатающее практически все виды полиграфической продукции. Мы провели на комбинате пять интенсивно заполненных и очень интересных дней, встречаясь с высококлассными профессионалами, и воодушевленные вернулись в Академгородок.

Однако, для того чтобы не утратить ориентации в этом многообразии, необходимо было выделить стержень производственной деятельности, доступный для целостного охвата, наиболее актуальный для применения вычислительных машин и в то же время находящийся в центре производственной деятельности комбината. Таким процессом нам представился выпуск «Правды» – газеты с жестким суточным циклом, проблемой отбора материала, требованием абсолютной достоверности, необходимостью постоянного ретроспективного анализа и обратной связи через письма читателей.

Это, однако, не означало, что процессы, относящиеся к другим подразделениям комбината, остались «за бортом». Наоборот, там, где связь с выпуском газеты носит органический характер – прежде всего набор, верстка и правка, – соответствующие подразделения и процессы изучались со всей возможной тщательностью.

Так все и получилось. «Системный анализ производственных процессов по выпуску газеты «Правда» был признан, причем обсуждение предложенных в нем принципов и гипотез активно происходило на разных уровнях, как научных, так и «наверху», «Проект РУБИН» был включен в план важнейших научно-исследовательских работ страны.

Генеральная схема создания и развития информационно-вычислительной системы РУБИН газеты «Правда», утвержденная в феврале 1979 года главной редакцией газеты в качестве основополагающего документа, определила «Информационно-Вычислительную Систему РУБИН» как «основанную на электронной вычислительной технике совокупность аппаратных, программных и организационно-методических средств, органически включаемых в процессы редакционно-издательской подготовки и выпуска газеты и имеющих своим назначением обеспечение комплексного справочно-информационного обслуживания редакции, автоматизированной обработки текстов при подготовке материалов, загрузки и поддержания информационных фондов, планирования и выпуска газеты».

Система РУБИН строилась как сочетание центрального вычислительного комплекса с активными, т.е. способными к автономному функционированию, периферийными терминалами. При этом реализация и внедрение системы начинается с ввода в строй ее периферийной части.

Разработка системы РУБИН была рассчитана на использование вычислительной техники стран СЭВ, допуская в целях обеспечения передового технического уровня применение элементов импортной техники в особых случаях. Ввод системы в эксплуатацию предусматривал постепенный переход к новым формам работы с полным учетом всего сложившегося опыта выпуска газеты и максимальным соответствием структуре редакции «Правды».

Главным видом работы при создании очередного номера газеты признавалась подготовка его текстовых материалов при их интенсивной многократной переработке. Для реализации указанных целей при разработке информационно-вычислительной системы генеральной схемой предусматривалось:

- Организация в системе двух отдельных потоков информации – информационно-справочного и текстового, замыкающихся соответственно на справочный фонд и архив текстов, и разделение справоч-

ного фонда и архива текстов на центральные фонды (для контрольно-справочных отделов и секретариата) и локальные фонды (ведущиеся в тематических отделах для их собственного использования).

- Хранение всех текстовых материалов (кроме оригиналов писем) в памяти ЭВМ и их автоматическая обработка по указанию пользователей посредством программных и аппаратных средств. Автоматизация набора материалов, находящихся в архиве текстов, с независимым заданием его полиграфического оформления, а также включение в систему средств, постоянной адаптации правил общения пользователей с системой.

Общая схема структуры системы РУБИН газеты «Правда» предполагала, что реализация функциональных возможностей системы осуществляется следующими средствами, распределенными по четырем основным группам.

1. Средства редактирования, которые обеспечивают эффективную и надежную обработку текстовых материалов при их многократной правке и замыкание всей технической стороны работы с текстами внутри ЭВМ, а также сервисное обслуживание работы с фондами;
2. Средства управления, обеспечивающие долгосрочное и оперативное планирование, подготовку очередных номеров, загрузки подразделений редакции и т.п.;
3. База Информации – для обеспечения справочно-информационного обслуживания при подготовке публикуемых материалов и проверке содержащихся в них фактов, создания и поддержания центральных и локальных информационных фондов и текстовых архивов, анализ почты газеты и ретроспективный анализ содержания газеты;
4. Средства набора, обеспечивающие прямое воспроизведение хранящихся в системе текстов.

Таково было мое видение развития системы весной 1979 года, выраженное в «Генеральной схеме», оптимистичное по функциональному наполнению и благожелательности сотрудников и консервативно-пессимистичное в отношении прогресса технических средств.

Разработки по проекту РУБИН начали осуществляться сразу в нескольких направлениях, как по периферии, так и по центральному вычислительному комплексу, второе направление работ возглавил в Москве проф. Э.З. Любимский. Однако непосредственный выход в редакцию по-прежнему определялся тем, какая аппаратура будет стоять на рабочих местах сотрудников. После долгих поисков и попыток разместить заказы в отечественной промышленности в проекте РУБИН в середине 1980 года появился новый адрес – г. Блоне под Варшавой, завод точного машиностроения «МЕРА-Блоне», который и стал нашим партнером по созданию периферии РУБИНа. Однако это уже другая история – история проекта МРАМОР.

### **Библиографический список**

1. Берс А.А. К анализу семантики базисных понятий информатики // Сборник трудов конференции, посв. 90-летию со дня рождения А.А. Ляпунова. Новосибирск, 2001.
2. Берс А.А. Строеие системы САПФИР // Новые задачи информатики / ВЦ СОАН. Новосибирск, 1979.
3. Берс А.А., Детушев В.А. Автоматизированный набор блочных текстов в системе САПФИР // Там же.
4. Берс А.А. Информационно-вычислительная система РУБИН газеты «Правда» // Прикладные методы информатики / ВЦ СОАН. Новосибирск, 1980.
5. Берс А.А. Программное обеспечение воспроизведения типографских шрифтов // Экспериментальная информатика / ВЦ СОАН. Новосибирск, 1981.
6. Берс А.А. От Гуттенберга к гипертексту // Шестые Макушинские чтения: сб. докл. Новосибирск, 2004.
7. Щедровицкий Г.П. Избранные труды. М., 1995.

## РАБОЧАЯ СТАНЦИЯ «МРАМОР»

**А.А. Берс**

Институт систем информатики им. А.П. Ершова СО РАН  
Новосибирск, РФ  
*baebers@iis.nsk.su*

Рассматривается опыт создания в 1980–1987 годах в рамках международного сотрудничества между СССР и ПНР для «Проекта РУБИН» газеты «ПРАВДА» рабочей станции «МРАМОР» – качественного рабочего места для издательской деятельности, построенного на слабой элементной базе. С советской стороны участвовали: Издательство «Правда», ВЦ СО АН СССР и Институт Прикладной математики им. М.В. Келдыша АН СССР, с польской стороны: завод точной механики «МЕРА-Блоне» и Центр телевизионной техники ЦОБРЕСПУ. Проект РУБИН был включен в состав программ важнейших научно-исследовательских работ ГКНТ СССР на 1980–1985 годы.

Архитектура, дизайн и функциональные характеристики РС МРАМОР и рабочих мест на ее основе, а также ее базовое и прикладное программное обеспечение разработаны в лаборатории академического института. Овеществлением этих идей в аппаратуре и подготовкой производства к серийному выпуску занималась конструкторская группа на заводе «МЕРА-Блоне». Большую поддержку этой работе в рамках «Проекта РУБИН» осуществляло Издательство «Правда».

В лаборатории экспериментальной информатики ВЦ СО АН СССР в работах по МРАМОРу участвовали: А.А. Берс, Ю.В. Бовкун, А.В. Коваленин, А.П. Мельник, А.Р. Муллағалиев, Г.П. Несговорова, Е.В. Овчаренко, В.Г. Поляков, С.Б. Руднев, М.В. Садовая, В.А. Четвернин.

Наши польские коллеги на заводе «МЕРА-Блоне»: М. Августиняк, Я. Завадский, Я. Заграек, М. Зюлковский, А. Колодеж, З. Лучук, Я. Матраш, Т. Мошевич, Р. Пацек и С. Шумский. Мониторы для РС МРАМОР созданы в Варшавском Центре телевизионной аппаратуры (ЦОБРЕСПУ) под руководством Е. Кани и Л. Непеклы.

Все текущие организационные трудности проекта легли на технический отдел Издательства «Правда» и на плечи его руководителя В.А. Тифенбаха.

Автор, главный конструктор разработки, считает своим приятным долгом отметить исключительно дружественное творческое взаимодействие всех членов этого международного коллектива из специалистов разного профиля. Я до сих пор искренне благодарен всем участникам этой интересной и долгой работы. Особо хочу и должен отметить то внимание и заботу, которое ей уделял наш Учитель – академик А.П. Ершов.

Организации проекта РУБИН предшествовали этапы обследования производственных процессов в редакции и издательстве «Правда» и системного анализа этих процессов. По результатам обследования «Генеральной схемой создания и развития системы РУБИН газеты "ПРАВДА", утвержденной главной редакцией «Правды» в феврале 1979 года, устанавливалось, что система будет состоять из периферийной сети и центрального вычислительного комплекса. ЦВК, с большой информационно-справочной базой данных для анализа и проверки публикуемых материалов и перспективного планирования выпусков газеты был реализован на старших моделях ЕС ЭВМ. Рабочая станция МРАМОР создавалась как терминальная база периферийной локальной сети рабочих мест для сотрудников редакции.

Было необходимо обеспечить автоматизацию трудоемких редакционно-издательских процессов с тем, чтобы большие объемы текстовой информации могли быть обработаны в жестко ограниченные сроки (что особенно важно при выпуске газет).

Цель работы состояла в создании программно-аппаратной базы, системы рабочих мест профессионалов – сотрудников редакции и издательства, поэтому разработка МРАМОРа с самого начала отличалась комплексным подходом к проблеме, совмещая проектирование аппаратуры, программного и шрифтового обеспечения и дизайн.

Была выпущена опытная серия из 40 рабочих мест, базовое программное обеспечение станции и прикладное программное обеспечение рабочих мест профессиональных полиграфических систем электронной подготовки изданий. Проводилась опытно-промышленная эксплуатация этих высокопроизводительных и удобных для пользователей рабочих мест, на которых осуществлено большое число изданий с высоким полиграфическим качеством (в том числе регулярный выпуск городской газеты «ЭНСК», подготовка ряда номеров журналов «Наука и жизнь» и «ЭКО» и большого числа книг).

Серийные экземпляры РС МРАМОР были экспонатами выставки «Сибирский прибор 87» (Академгородок, 1987), где разработка была удостоена Диплома за оригинальное решение, Всепольской конференции «POLIKON-87» (Познань, 1987) и выставок «Завод "Мера-Блоне" советской информатике» (Москва, 1997; Вильнюс, 1988).

На конкурсе прикладных работ СО АН СССР 1988 года рабочей станции МРАМОР был присужден Диплом III степени.

Проведенное в 1977 году обследование издательства и типографии газеты «Правда», по результатам которого был подготовлен отчет «Системный анализ производственных процессов по выпуску газеты "Правда"», позволило выделить три основных канала переработки информации и их существенные характеристики, необходимые для создания системы РУБИН:

- обработка текстов, набор, верстка и подготовка печатных форм газетных полос,
- планирование номеров и управление выпуском газеты,
- справочно-информационная служба и проверка публикуемых фактов.

Различные рабочие места сотрудников редакции, издательства и типографии должны были обеспечить совместный согласованный выход ко всем этим трем потокам, реализацию в разных сочетаниях (определяемых типом рабочего места) требуемых прикладных функций и возможность использования результатов одних работ в ходе других.

При реализации проекта РУБИН было необходимо учесть ряд обязательных ограничений, диктуемых заказчиком и обстановкой:

- в основном аппаратура должна была быть построена только на отечественной элементной базе и компонентах, в крайних случаях разрешалось применение микросхем, компонентов и устройств, производимых в странах-участниках СЭВ;
- требовалось в максимальной степени сохранить сложившуюся в редакции технологию работы, стиль и принципы взаимодействия ее сотрудников, не потеряв при переходе на ЭВМ привычной для них комфортности рабочего места, при этом не могло быть и речи о том, что пользователи такого социального уровня будут согласны приспосабливаться к неудобным для них функциям и особенностям аппаратуры и программ.

С другой стороны, следует вспомнить, что в конце 70-х годов основной парк ЭВМ нашей страны был составлен системами ЕС и СМ ЭВМ, оснащенными главным образом заимствованным, как говорилось, «Штатным» программным обеспечением. По своим характеристикам и конструкции они никак не были приспособлены для работы в условиях редакции. Терминальное оборудование, пригодное для редакционно-

издательской подготовки полиграфических изданий практически отсутствовало, выпускаемый в стране комплекс фотонаборных автоматов (Каскад) с носителями на перфоленте уже был морально устаревшим, а шрифтовое обеспечение фотонабора и терминалов – просто примитивным.

Эксплуатируемые зарубежные типографские системы на ЭВМ плохо соответствовали требованиям отечественной практики и затрудняли выпуск изданий на русском и других языках народов СССР.

Если для центрального вычислительного комплекса можно было выбрать одну из старших моделей ЕС ЭВМ и создавать прикладные программы, опираясь на существующие операционные системы и СУБД, то для терминального уровня системы РУБИН было очевидно, что на существующей в тот момент технике решить проблему автоматизированного рабочего места только посредством создания программного обеспечения для какой-либо из фактически доступных мини- или микроЭВМ не удастся. Поэтому и было принято решение о создании новой машины класса макро-мини, получившей название «Рабочая станция МРАМОР».

Исходя из доступной элементной базы и конструктивов было ясно, что вычислительных мощностей одного микропроцессора типа K580BM80 будет недостаточно для обеспечения требуемых функций и свойств рабочих мест.

Поэтому возникшая перед нами проблема решалась совместной разработкой аппаратуры и программного обеспечения нового программно-аппаратного комплекса, при создании которого необходимо было, кроме того, исходить из конкретных возможностей найденного с большим трудом завода-изготовителя (Польского завода точной механики «МЕРА-Блоне»).

При разработке РС МРАМОР в проект закладывались следующие требования к возможностям нового программно-аппаратного комплекса:

- модульность аппаратуры, базового и прикладного программного обеспечения;
- многопроцессорность, включая возможную гетерогенность аппаратуры и программных средств;
- многозадачность обслуживания пользователя при мультипрограммной загрузке процессоров задачами;
- множественность рабочей станции со связью между рабочими местами;
- многооконность интерфейса и, в том числе, возможность многоэкранности рабочего места;
- федеративность, т.е. оперативное взаимодействие, позволяющее пользователям, процессам или процессорам работать в коллективном режиме;
- высокое качество представления на экране и в рабочих распечатках полиграфических аспектов обрабатываемых изданий, в том числе разнообразия национальных языков, типов письменности и шрифтов;
- перестраиваемость конфигурации рабочего места, как по составу аппаратуры и программного обеспечения, так и по дизайну;
- координация доступа к общим для редакции базам данных, находящимся в центральной ЭВМ и работы на персональном рабочем месте.

Чтобы все эти разнообразные требования можно было реализовать в некотором целостном подходе, объединяющем возможности аппаратных и программных средств и позволяющем обоснованно принимать проектные решения, была разработана концептуальная модель организации исполнения вычислительных процессов. Кроме того, с учетом особенностей подготовки разнообразных изданий в издательстве «Правда» и опыта, полученного при разработке системы САПФИР для Первой образцовой типографии, была сформирована концептуальная модель обеспечения электронной подготовки изданий.

Эти концептуальные модели вместе с перечисленными выше требованиями были положены в основу разработки РС МРАМОР, в ходе которой удалось решить ряд новых для того времени задач:

- практически осуществить совместное проектирование единой целостной программно-аппаратной обстановки, включающей аппаратуру, базовое и прикладное программное обеспечение, интерфейс пользователя и дизайн рабочего места;
- разработать оригинальную открытую аппаратуру с гетерогенными вычислительными блоками и обеспечить требуемые характеристики рабочих мест, даже при слабости доступной элементной базы, посредством компенсации этой слабости за счет архитектурных и конструктивных решений путем возможности выбора между аппаратной и программной реализацией необходимых функций;
- создать модель организации вычислений – «Операционные обстановки высокого уровня», позволившую организовать совместное исполнение в виде рабочей смеси большого числа взаимодействующих процессов, протекающих в интересах нескольких пользователей на множестве виртуальных (в том числе реальных) процессоров, каждый из которых работает в мультипрограммном режиме;
- разработать операционную систему и базовое программное обеспечение, управляющие работой многопроцессорного комплекса на основе данной модели вычислений;
- создать многопроцессорную систему из разнородных компонент и организовать их совместную работу в режиме мультипрограммного исполнения большого числа связанных процессов в интересах нескольких пользователей одновременно, обеспечивающую реализацию достаточной степени параллелизма работы посредством организации рабочей смеси виртуальных машин на гетерогенных вычислительных средствах;
- создать возможности для дальнейшего расширения функций РС МРАМОР за счет пополнения состава ее модулей, устройств и, главным образом, прикладного программного обеспечения и получить большое разнообразие реализуемых на РС МРАМОР конфигураций рабочих мест при малом числе типов производимых модулей.

Таким образом, в МРАМОРе был осуществлен ряд оригинальных архитектурных, конструктивных и программных решений, некоторые из которых в настоящее время стали обыденными (многошинная архитектура с разнородными процессорами, портретное положение дисплея, многооконный интерфейс, аккордный ввод с клавиатуры, программируемые шрифты с различными начертаниями знаков переменной ширины).

В составе опытной серии были выпущены две модели РС МРАМОР: двухместная и четырехместная. Кроме того, для нужд разработчиков были скомпонованы два экземпляра одноместной полиэкранной рабочей станции.

Вычислительные блоки станции построены на микропроцессоре типа 8086 и его сопроцессорах: арифметическом и ввода-вывода. Блок реализует связи между системной и локальной шинами, содержит схемы для 15 входов прерываний, к любому из которых можно присоединить любой из источников прерывания, механизм страничной адресации, микросхемы СПЗУ с общим объемом до 32 К 16-разрядных слов.

Для вычислительных блоков был разработан оригинальный механизм расширения адресации: схема, определяющая по сигналам состояния и порядковому номеру цикла в команде, зачем процессор обращается в память. К выданному процессором 16-разрядному адресу выборки команды, работы со стеком или работы с данными приписывался один из трех двухразрядных префиксов, задаваемых программно и хранящихся в специальном регистре. Таким образом, оперативная память разбивается на четыре подпространства, любые три из которых могли одновременно быть назначены программе в качестве подпространств команд, данных и стека.

Обеспечен страничный доступ к памяти с помощью аппаратно реализованной таблицы переадресации (ТПА). Любое слово ТПА может быть непосредственно прочитано и записано процессором. Предусмотрены два варианта размера страницы в 4 Кб и 16 Кб. Из ТПА выдаются сигналы, направляющие адрес либо в

ПЗУ, либо на свою ЛШ, либо на СШ (а через нее и на ЛШ другого блока). В варианте с 16 Кб используются все биты процессорного адреса, причем 14 младших из них определяют смещение на странице.

В результате процессор имеет одновременно доступ к 64 страницам, каждая из которых может находиться в одном из независимых адресных подпространств памяти: Системной Шины, Локальной Шины любого из вычислительных блоков и ПЗУ данного блока. Логический диапазон адресации СШ – 16 М, каждой из ЛШ – 1М, ПЗУ вычислительного блока – 64 К. Доступ каждого процессора через СШ на локальную шину любого другого вычислительного блока имеет приоритет перед доступом от местного процессора, поэтому если процессор выдает сигнал блокировки (Lock) на Системную шину, то это препятствует выходу на нее других вычислительных блоков и правильная работа с командами типа **test-and-set** (семафорами) обеспечивается даже при их размещении в локальной памяти.

Операционная система ОНИКС позволила использовать в режиме «коллективного подряда» несколько процессоров разного типа и обеспечивала «смешанное исполнение» рабочей смеси для нескольких языков – ЯВА (Язык вместо ассемблера), СИДУЛА (Си + Модула) и ФОДУЛА (ФОРТ+Модула).

Рабочие места на базе МРАМОРа позволили осуществлять разработку программного обеспечения, создания шрифтов и электронную подготовку издания для ряда газет и книг, в том числе, например, русско-китайского разговорника.

В одной из своих последних заметок А.П. Ершов [*Смотреть вперед, видеть по сторонам. «Колонка редактора»* // *Микропроцессорные средства и системы. 1988. № 2*] выявил прямой параллелизм между американским прогнозным перечнем (проф. У. Бим) основных характеристик персональных ЭВМ для 90-х годов и возможностями, воплощенными в архитектуре опытной серии РС МРАМОР.

Много внимания при создании РС МРАМОР было уделено общей компоновке и дизайну рабочих мест. В отличие от общепринятого подхода, когда персональная ЭВМ, ее дисплей и другие блоки размещаются на столе, занимая почти все место на нем, при проектировании РС МРАМОР было отдано предпочтение «организации большого свободного рабочего стола». Поэтому все модули и НГМД размещены в приставной тумбе стола, снабженной роликами, для клавиатуры предусмотрена опорная рамка, выдвигающаяся из-под поверхности стола и позволяющая обеспечить правильную высоту клавиатуры и регулировать угол ее наклона. Клавиатура имеет 137 клавиш; значение 134 из них – полностью перепрограммируемо, т.е. их раскладка и семантика полностью определяются программным драйвером.

Клавиши на клавиатуре сгруппированы в шесть эргономических полей, три из которых являются полями функциональных клавиш, одно соответствует клавиатуре пишущей машинки и еще два предназначены для числового ввода и



управления курсором. Пять регистровых клавиш могут обеспечить работу в 32 регистрах по 128 символов в каждом, позволяя использовать в одном издании до 4096 разных символов. Поля клавиш выделены цветом и разделены промежутками, позволяющими разместить наклейки с обозначениями функций. Для сигнализации пользователю от программ на клавиатуре есть 8 светодиодов красного, желтого и зеленого цветов, три из них совмещены с фиксируемыми в нажатом положении клавишами.

Для МРАМОРа были сделаны блоки прямого вывода на фотонаборную часть автомата ФА-1000 и фотоприемное устройство «Газета-2», что обеспечило получение качественных форм офсетной печати.

В РС МРАМОР осуществлено программное управление выключением сетевого питания. Это обусловлено тем, что отдельные рабочие места станции могут находиться в соседних комнатах. Закончив работу, пользователь может отключить лишь устройства своего рабочего места (монитор и дисководы), но не станцию в целом. Все остальные заботы переложены на базовое программное обеспечение.

Для рабочих мест сотрудников редакции важнейшим параметром, определяющим удобство и пригодность их к эксплуатации, является качество показа текстовой информации, обеспечивающее представление большого ассортимента символов и разнообразия их выделений. Поэтому для МРАМОРа был заказан специальный монохромный монитор с диагональю 50 см и повышенным временем послесвечения (для устранения мерцания на экране при стандартной телевизионной развертке). Кроме того, было признано негибким использование ПЗУ для генерации знаков и выбрана схема дисплея с опорной памятью, а для лучшего использования экрана при соответствии изображения на нем формату печатных полос было принято расположение экрана с вертикальной ориентацией его длинной стороны (называемое «портретным», в отличие от «ландшафтного», применяемого в телевизорах).

При этом сохранена развертка изображения по ТВ-стандарту, т.е. по вертикали, поперек текстовых строк. ОЗУ изображения в 64 Кб позволило получить поле 768 x 576 точек с тремя уровнями яркости. Монитор подвешен над столом на специальном поворотном кронштейне, высоту расположения экрана и его наклон можно регулировать.

### **Система подготовка изданий**

На РС МРАМОР реализована профессиональная система электронной подготовки полиграфических изданий различных классов и произвольной сложности, обеспечивающая издательству полный технологический цикл от первичного ввода текстов до верстки издания и выдачи комплекта готовых фотоформ.

Система опирается на принцип отделения текста издания от его полиграфического оформления. Собственно текст в системе представляется двойной последовательностью. Ее старший уровень образован разнотипными элементами издания, доступ к которым осуществляется через рабочее оглавление, содержащее также типы и внешние характеристики элементов.

Полиграфическое оформление издания соотнесено с типами элементов издания и/или разметкой в тексте, а также представлено отдельной пополосной структурой. Внутритекстовые ссылки и сноски реализуются указателями, включаемыми в указанные структуры.

Диалоговая разработка макета издания и его пополосная верстка ведутся по рабочему оглавлению, редактирование и корректура поддерживаются системой взаимосвязанных окон.

Шрифтовые выделения представлены специальными экранными шрифтами, но расчет форматов строк ведется по ширинам реальных наборных шрифтов.

Разработаны технологии и средства электронной подготовки изданий за счет отделения собственно текста от его полиграфического исполнения и возможности их отдельной обработки; разработано совме-

стное применение структурно-ссылочного представления корпуса текстов в сочетании с текстовыми разметками для обеспечения эффективного проведения глобальной обработки текста (в том числе макетирования и верстки); определена номенклатура рабочих мест различных подразделений редакции и возможность динамического перехода в процессе работы на одном физическом месте к функциям разных логических рабочих мест.

Обеспечена возможность подготовки смешанных изданий на различных национальных языках и разных типах письменности в рамках одного рабочего места.

### Библиографический список

5. Берс А.А. Информационно-вычислительная система РУБИН газеты «Правда» // Прикладные методы информатики. Новосибирск, 1980.
6. Берс А.А. Программное обеспечение воспроизведения типографских шрифтов // Экспериментальная информатика. Новосибирск, 1980.
7. Берс А.А., Поляков В.Г., Руднев С.Б. О системе программирования высокого уровня со смешанными вычислениями для персональных микропроцессорных комплексов // Актуальные проблемы развития архитектуры и программного обеспечения ЭВМ и вычислительных систем. Новосибирск, 1983.
8. Берс А.А., Поляков В.Г. Архитектура многофункционального автоматизированного рабочего места обслуживания редакции // Персональные ЭВМ в задачах информатики. Новосибирск, 1984.
9. Берс А.А., Поляков В.Г. Особенности системного программного обеспечения многофункционального автоматизированного рабочего места обслуживания редакции // Там же.
10. Берс А.А. Рабочая станция нового поколения МРАМОР // Разработка ЭВМ нового поколения: Архитектура, программирование, интеллектуализация. Новосибирск, 1986.
11. Берс А.А. Об объектной ориентации и организации архитектуры программных систем // Актуальные вопросы технологии программирования. Л., 1989.

По адресу <http://ershov.iis.nsk.su/russian/> доступны полные тексты материалов, хранящиеся в архиве А.П. Ершова (раздел Программные проекты: МРАМОР, РУБИН):

1. Системный анализ производственных процессов по выпуску газеты «Правда» // Совместный отчет ВЦ СО АН СССР и НФ ИТМиВТ АН СССР. Новосибирск, 1977.
2. Берс А.А. Генеральная схема создания и развития информационно-вычислительной системы РУБИН газеты «Правда». М.; Новосибирск, 1979.

## ИСТОРИЯ СОЗДАНИЯ И ДЕЯТЕЛЬНОСТИ МИНСКОГО НАУЧНО-УЧЕБНОГО ЦЕНТРА «АЛГОРИТМ»

**Н.Ф. Богданова**

Институт подготовки научных кадров НАН Беларуси

Минск, Беларусь

[nina@kint.basnet.by](mailto:nina@kint.basnet.by)

Минский научно-учебный центр «Алгоритм», преобразованный из курсов заказчиков при Минском заводе ЭВМ им. Орджоникидзе, входил в систему учебных центров Государственного комитета по вы-

числительной технике и информатике СССР. Такие центры осуществляли подготовку и повышение квалификации специалистов по техническим и программным средствам вычислительной техники для различных отраслей.

Отделы, занимавшиеся подготовкой слушателей по техническому обеспечению, специализировались практически по каждому устройству ЭВМ серии ЕС.

Отдел подготовки по центральным устройствам состоял из секторов, специализировавшихся по конкретным типам ЭВМ (вначале это были сектора по ЭВМ ЕС 1022, ЕС 1035, ЕС 1060). Впоследствии, по мере выпуска новых ЭВМ серии ЕС, тематика работы секторов менялась. Центральные устройства делились на несколько основных блоков, в соответствии с названиями которых давались и названия предлагаемых для прохождения обучения специализаций. Так, в ЕС 1060 и ЕС 1061 процессор состоял из блоков центрального управления (ЦУ), арифметико-логического устройства (АЛУ), блока управления памятью (УП) и блока контроля и диагностики (КД). Соответственно, готовили по двум специализациям: ЦУ – АЛУ (здесь же читался и блок КД) и УП.

Вначале читался общий курс. При проведении спецкурса (по блокам центрального процессора) проводилась обязательная практика на соответствующих ЭВМ. Слушатели должны были запустить диагностические тесты и найти неисправность. Практические занятия часто проводились по вечерам, поэтому случалось, что уставший преподаватель забывал снять перемычку, имитировавшую неисправность. Работники ВЦ так привыкли к этому, что поиск любой неисправности всегда начинали с поиска оставленной перемычки.

После окончания курсов слушателям выдавались удостоверения единого для СССР образца. Слушатели МНУЦ «Алгоритм», работники ВЦ, приезжали со всего Союза: от Москвы до Владивостока и Хабаровска. Многие из них уже проходили подготовку или здесь же, в Минском центре, или в аналогичных центрах страны (Пенза, Казань и т.д.). При центре была построена гостиница для проживания слушателей.

В 1991 году центр распался на множество малых предприятий. Одним из которых стало Государственное научно-учебное малое предприятие «Алгоритм ПС», тоже занимавшееся обучением пользователей ЭВМ, но уже персональных.

## КАНДИДАТСКИЙ МИНИМУМ ПО ИНФОРМАТИКЕ: ИСТОРИЯ И СОВРЕМЕННОСТЬ (1987–2006)

**И.Ф. Богданова**

Институт подготовки научных кадров НАН Беларуси

Минск, Беларусь

*bogdanova@kivt.basnet.by*

Распоряжением Президиума АН СССР от 13 июня 1986 года № 10105–1073 был введен новый обязательный кандидатский минимум по информатике и вычислительной технике. Для подготовки аспирантов и приема у них кандидатского дифференцированного зачета по новому предмету во всех академиях наук союзных республик были созданы соответствующие кафедры. В Академии наук БССР летом 1987 года тоже была создана кафедра информатики и вычислительной техники (КИВТ), с осени того же года она начала осуществлять возложенные на нее функции.

Обучение проводилось по программам ВАК СССР. С 1988 года программы были усложнены и состояли уже из двух разделов, которые изучались два семестра, за 100 аудиторных часов. С 1996 года кандидатский минимум по информатике и вычислительной технике ВАК РБ был преобразован в кандидатский минимум по «Основам информационных технологий» с новой типовой программой-минимум. С 1998 года этот кандидатский минимум сдается в объеме, состоящем из двух частей: типовой программы-минимум и дополнительной программы, отражающей современные тенденции и специфику применения современных информационных технологий в области научной деятельности аспиранта. В декабре 2004 года типовая программа-минимум была модернизирована.

С 1987 года и по сегодняшний день после окончания аудиторных занятий аспиранты выполняют выпускную работу исследовательского характера.

История обучения аспирантов белорусской Академии наук на кафедре информатики и вычислительной техники полностью совпадает с историей создания и развития отечественной вычислительной техники соответствующего периода. Вначале кафедра не имела собственных компьютерных классов и обучение проводилось в научных институтах АН БССР, располагавших большими вычислительными центрами с достаточным количеством терминалов на больших ЭВМ ЕС-1035, ЕС-1046, и ЕС-1060 и малых ЭВМ СМ-4, а также на ПЭВМ Искра-226, МЕРА-125.

С 1991/92 учебного года все аспиранты обучаются уже только в компьютерных классах КИВТ. За прошедшие годы эти классы переоснащались следующими компьютерами: Нейрон, РОБОТРОН, Мера-125, Электроника-60, ДВК-2, ЕС 1840, ЕС 1841, ЕС 1842, Искра-226, Искра-1030, Искра-1030М, IBM PC 286, 386, 486, Pentium 100, Pentium IV.

За прошедшее время аспиранты изучали такие языки программирования, как Бейсик, Фортран, Паскаль, С, Дельфи.

## ПРОЕКТ WEB-SYNDIC: СИСТЕМА УДАЛЕННОГО РЕШЕНИЯ ЛИНЕЙНЫХ ДИОФАНТОВЫХ УРАВНЕНИЙ В НЕОТРИЦАТЕЛЬНЫХ ЦЕЛЫХ ЧИСЛАХ

**Ю.А. Богоявленский, Д.Ж. Корзун, К.А. Кулаков, М.А. Крышень**

Петрозаводский государственный университет

Петрозаводск, РФ

*ybgg@cs.karelia.ru*

### Введение

Задача численного решения одНЛДУ представляет практический интерес для ряда приложений, включающих, например, системы искусственного интеллекта [1], оптимизацию кода программ [2, 3], исследование операций [4] и дискретное моделирование сетевых систем [5, 6]. В силу вычислительной сложности задачи решения системы одНЛДУ возникает проблема выбора подходящей реализации алгоритма решения (решателя) при разработке соответствующего программного обеспечения (ПО).

Теоретические оценки сложности не всегда дают всю необходимую информацию о поведении решателя для заданного класса систем. Например, алгоритмы, имеющие одинаковые теоретические оценки сложности, могут по-разному вести себя на практике, равно как и различные реализации одного и того же алгоритма. Более того, решатель, как и любая программа, не свободен от ошибок реализации.

Перечисленные выше проблемы приводят к необходимости использования некоторой экспериментальной площадки для исследования решателей. Пользователю демонстрируется работа решателя на тестовых и эталонных примерах систем, включая сгенерированные автоматически или вводимые вручную. Каждый запуск решателя является тестом, т.е. выполняется неявное тестирование решателя, включая проверку полученного решения (автоматическая или непосредственно пользователем). Возможность измерения показателей производительности одного или нескольких решателей используется для разработки рекомендаций по их применению.

Цель проекта Web-SynDic заключается в разработке такой системы для частного класса систем одНЛДУ, ассоциированных с контекстно-свободными грамматиками (системы одАНЛДУ). Для таких систем известны эффективные синтаксические алгоритмы решения [7, 8, 9]. Выбор данного класса систем обусловлен его важностью для задач дискретного моделирования. В то же время экспериментальная сложность синтаксических алгоритмов требует дополнительных исследований.

Проект Web-SynDic является научно-исследовательским студенческим командным проектом (<http://websyndic.cs.karelia.ru/>). Его успешный опыт еще раз показывает перспективность включения подобной формы обучения в учебные планы подготовки высококвалифицированных специалистов в области информатики и программирования [10].

Статья имеет следующую структуру. В п. 1 дано введение в предметную область систем одАНЛДУ, поставлены задачи решения и генерации тестовых и эталонных систем, дан перечень используемых решателей и генераторов. В п. 2 описаны основные функции системы Web-SynDic с точки зрения пользователя. В п. 3 дана архитектура программной системы в целом и ее важнейшей части, ответственной за работу решателей и генераторов. В п. 4 представлен интерфейс пользователя и обсуждены типичные действия пользователя. Процесс разработки системы Web-SynDic в рамках студенческого командного проекта описан в п. 5. Краткий обзор уже выполненных экспериментальных исследований представлен в п. 6.

## 1. Задачи решения и генерации систем одАНЛДУ

Базис Гильберта системы одНЛДУ есть множество всех неразложимых ее решений, где под неразложимостью понимается невозможность представления решения в виде суммы двух ненулевых решений [4]. Мы рассматриваем следующие задачи численного решения систем одНЛДУ: 1) поиск частного ненулевого решения и 2) нахождение базиса Гильберта. Существуют различные алгоритмы решения систем одНЛДУ (см. напр.: [11, 12, 13]) и соответствующие им решатели. Задача поиска частного решения имеет полиномиальную сложность [14]. В то же время алгоритмы нахождения базиса Гильберта носят переборный характер. Известным нам исключением является синтаксический алгоритм, который является псевдополиномиальным, – сложность  $O(Q^3 m^2 n)$ , где  $Q$  – константа, ограничивающая число базисных элементов [9],  $m$  и  $n$  – число неизвестных и уравнений в системе соответственно. Однако класс систем одНЛДУ, решаемых алгоритмом, ограничен ассоциированными с контекстно-свободными грамматиками системами (системы одАНЛДУ) [7, 8]. Алгоритм реализован в виде решателя syntactic solver [9]. Для решения систем одАНЛДУ можно также применять и другие решатели.

Решатель, вообще говоря, характеризуется классом задач, для которых возможно его практическое применение. Область применения решателя ограничивают параметрами решения. Для характеристики класса используют ограничения на структуру систем одНЛДУ, на величину ее коэффициентов и на число базисных решений. С точки зрения потребления решателем вычислительных ресурсов используют такие

параметры решения, как максимально допустимое время и память. Таким образом, возникает задача оценки конкретного решателя на основе заданных параметров решения.

Для экспериментальной оценки решателя syntactic solver и его сравнения с другими решателями предлагается подход на основе автоматической генерации тестовых и эталонных систем одАНЛДУ [15]. Тестовые системы одАНЛДУ предназначены для оценки корректности решателя, включая варианты массового и стрессового тестирования. Эталонные системы предназначены для характеристики решаемого класса задач и сравнительного анализа с альтернативными решателями на основе измеренных показателей производительности. В обоих случаях задача генерации требует построения как самой системы одАНЛДУ, так и ее решения (базис Гильберта). Кроме того, автоматическая генерация позволяет эффективно демонстрировать возможности решателя. Автоматическая генерация регулируется такими параметрами, как число уравнений и неизвестных в системе одАНЛДУ, максимальный размер ее коэффициентов, максимальный размер базиса Гильберта, максимально допустимое время на генерацию и др.

В рамках проекта Web-SynDic помимо основного решателя syntactic solver были подключены следующие альтернативные решатели:

- Решатель slopes [12]. Реализует алгоритм нахождения базиса Гильберта произвольной системы одАНЛДУ, основанный на геометрических свойствах множества решений.
- Решатель lp\_solve [16]. Реализует алгоритм решения задачи целочисленного линейного программирования (ЦЛП) на основе симплекс-метода и метода ветвей и границ. Позволяет находить частное решение системы одАНЛДУ как оптимальное решение для некоторой целевой функции с ограничениями в виде исходной системы одАНЛДУ.
- Решатель GLPK [17]. Пакет программ решения задач линейного программирования на основе симплекс-метода или метода внутренних точек. Для решения задач ЦЛП используется метод ветвей и границ.

Для автоматической генерации тестовых и эталонных систем одАНЛДУ были реализованы и подключены к Web-SynDic три алгоритма генерации:

- Генератор Jordan. Представляет собой аналог преобразования Гаусса – Жордано. Генерирует систему одАНЛДУ с базисом Гильберта, состоящим из единичных векторов.
- Генератор Gauss. Представляет собой аналог преобразования Гаусса. Генерирует систему одАНЛДУ с базисом Гильберта, основанном на единичных векторах.
- Генератор ExtGauss. Расширяет алгоритм решателя Gauss, позволяя генерировать системы с большим числом нулевых коэффициентов системы и более сложной структурой базиса Гильберта.

Работа над новыми генераторами продолжается, обзор ее текущего состояния можно найти в [18].

## 2. Функции системы Web-SynDic

*Работа с одиночной системой одАНЛДУ (базовая функция).* Система Web-SynDic позволяет автоматически генерировать систему одАНЛДУ с помощью подключенных генераторов или вводить систему вручную. Полученную систему одАНЛДУ можно решить с помощью syntactic solver. В результате предоставляется отчет о решении, который содержит исходную систему одАНЛДУ, показатели производительности решателя (время и память), характеристики системного окружения (аппаратное и программное обеспечение) и вычисленный базис Гильберта.

Решатель syntactic solver является основным и всегда используется для решения. В качестве дополнительного варианта можно выбрать один из альтернативных решателей и выполнить сравнительный анализ.

Для альтернативного решателя также вычисляются показатели производительности. Решение (частное решение или базис Гильберта), найденное альтернативным решателем, сравнивается с базисом Гильберта, полученным syntactic solver. В результате возможны следующие ситуации: 1) решения не противоречат друг другу, 2) решения различаются и 3) один из решателей (или оба) не смог найти решение в рамках заданных ограничений на максимально допустимые время и память.

Данная функция демонстрирует базовые возможности синтаксического алгоритма, включая сравнение с альтернативными решателями. В дальнейшем предполагается расширение этой функции для выполнения тестирования, что включает проверку найденных решений на корректность (непосредственная подстановка в систему) и сравнение с эталонным решением, предоставляемым генератором. В настоящее время эти расширения реализованы, но не подключены к основной версии системы Web-SynDic [15].

*Работа с множеством систем одАНЛДУ.* Расширяет предыдущую функцию, позволяя автоматически генерировать или вводить вручную некоторый класс систем одАНЛДУ. Данная функция позволяет исследовать производительность решателя syntactic solver для заданного класса систем, включая сравнительный анализ с одним из альтернативных решателей.

В отчете о решении множества систем одАНЛДУ содержится краткая характеристика исходного множества систем (минимальный, средний и максимальный коэффициенты, минимальное, среднее и максимальное число базисных решений и др.). Вычисляются такие показатели производительности, как суммарное время, затраченное на решение всех систем из множества, и максимальный объем памяти среди всех решаемых систем. В текущей версии Web-SynDic найденные решения, равно как и сами системы, в отчете не предоставляются.

Системы из множества решаются в порядке их описания. Если все системы множества успешно решены, то формируется отчет об успешном решении. Если достигнута система, которая не смогла быть решена, то работа с множеством завершается. В этом случае считается, что решатель не справился с предложенным множеством систем, а в качестве показателей производительности в отчет выводятся значения для усеченного множества, состоящего из текущей системы (нерешенной) и всех ей предшествующих.

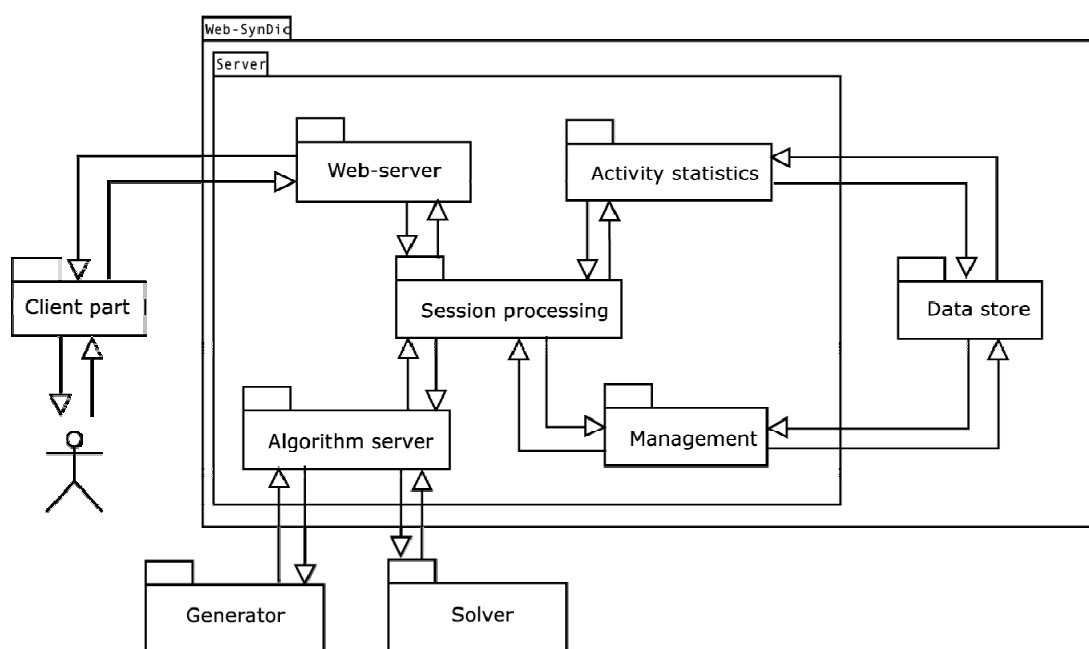
*Регистрация и учет пользователей.* Идентификация пользователей происходит посредством входного имени и пароля пользователя. Зарегистрированные пользователи могут использовать обратную связь, иметь свои персональные настройки (сохраняются в системе) и получить статус привилегированного пользователя. Для последнего отсутствуют ограничения на максимальные значения параметров решения и генерации.

Привилегированный пользователь может выступать в качестве администратора системы с правом изменения персональных данных любого пользователя, установки максимальных значений параметров решения и генерации для анонимных и зарегистрированных пользователей. Администратору также доступна информация об использовании системы. В частности, возможен просмотр таких показателей использования, как количество сессий, количество запросов на генерацию/решение, время работы и ряд других.

*Конфигурация параметров решения и генерации.* Пользователи могут конфигурировать такие параметры, как максимально допустимые время решения и объем памяти, верхние границы коэффициентов системы одАНЛДУ, максимальный размер базиса Гильберта и ряд других. Непривилегированный пользователь не может превышать максимальные значения параметров решения и генерации. Для анонимных пользователей настройки сохраняются только на время работы сессии.

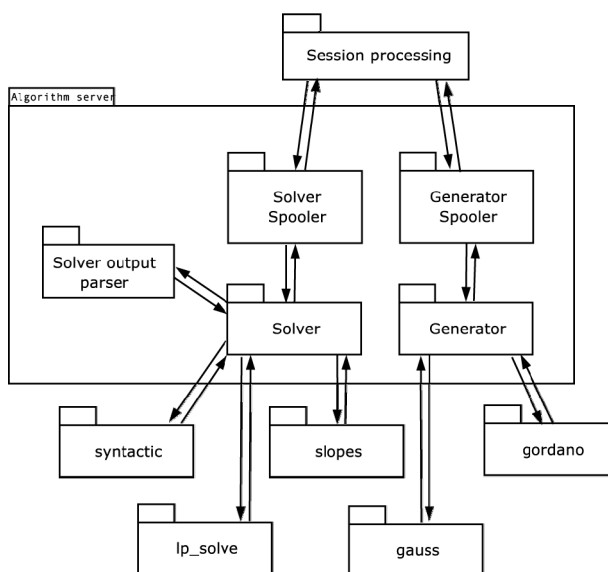
## Архитектура системы Web-SynDic

Система Web-SynDic реализована на основе архитектуры клиент-сервер (рис. 1) [19].



**Рис. 1. Высокоуровневая архитектура системы Web-SynDic**

В качестве клиента пользователь использует стандартный web-обозреватель. Работа пользователей с системой Web-SynDic происходит в режиме сессий. Все запросы поступают в подсистему Web-server, которая переводит данные во внутренний формат и передает их подсистеме Session processing. Последняя обеспечивает связь между остальными подсистемами сервера и разделяет обработку запросов по сессиям. Подсистема Activity statistics накапливает статистику об использовании системы в целом и предоставляет отчеты администратору. Подсистема Management управляет настройками и ограничениями пользователя. Обе подсистемы Activity statistics и Management работают с хранилищем данных Data store.



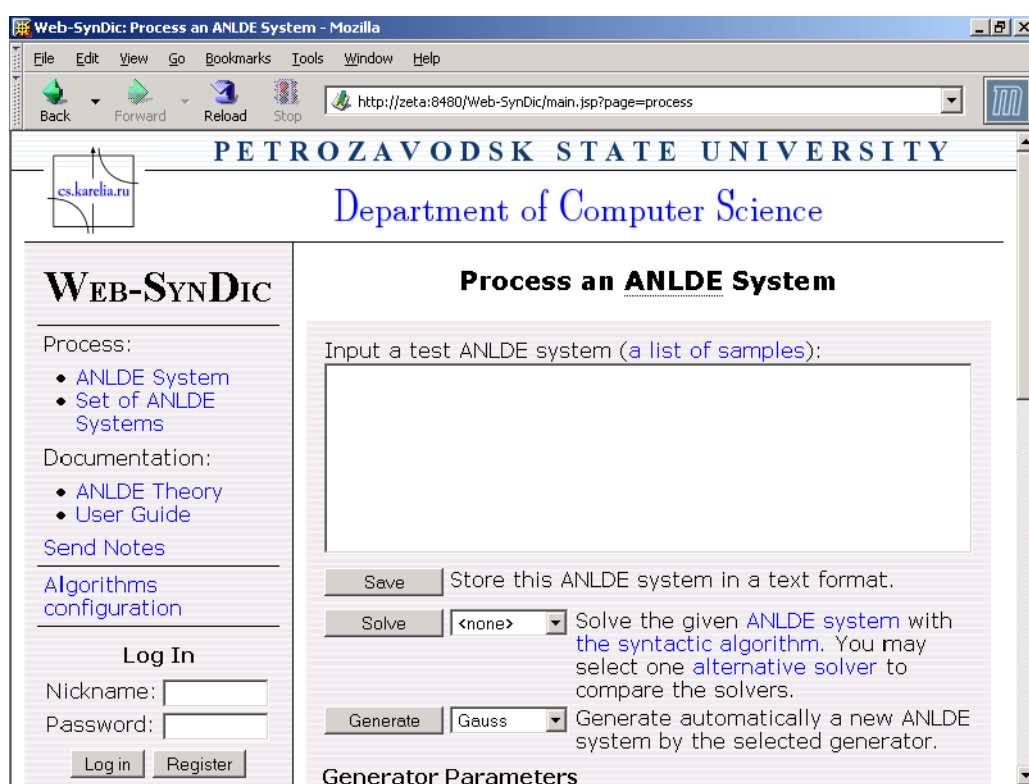
**Рис. 2. Архитектура подсистемы Algorithm server**

Решатели и генераторы являются внешними объектами по отношению к системе Web-SynDic. Подсистема Algorithm server обеспечивает взаимодействие системы Web-SynDic с решателями и генераторами, предоставляя пользователю доступ только к результатам их работы. Архитектура подсистемы Algorithm server представлена на рис. 2 [19].

Модули Solver spooler и Generator spooler управляют выполнением задач генерации и решения. Модули Solver и Generator обеспечивают интерфейс с решателями и генераторами. Модуль Solver output parser отвечает за анализ результатов работы алгоритмов решения и перевод их во внутренний формат. Измерение параметров производительности использует разработанную ранее программную систему alg\_analyzer [15].

### Интерфейс пользователя

Интерфейс пользователя реализован в среде Java в виде набора документов JSP и сервлетов. Фрагмент одного из окон представлен на рис. 3.



**Рис. 3. Фрагмент окна для работы с одиночной системой одАНЛДУ**

Окна интерфейса разделены на две части: левая представляет собой меню быстрого доступа к функциям системы, а правая содержит справочную информацию, форму ввода данных или отчет о результатах работы.

Меню быстрого доступа зависит от типа пользователя (анонимный пользователь, зарегистрированный или администратор). Любому пользователю доступны следующие пункты:

- Работа с одиночной системой одАНЛДУ (Process ANLDE system).
- Работа с множеством систем одАНЛДУ (Process set of ANLDE systems).
- Библиографический список литературы по теории систем одАНЛДУ (ANLDE Theory).

- Руководство пользователя (User Guide).
- Настройки алгоритмов генерации и решения (Algorithms configuration).

Справочная информация включает руководство пользователя, библиографический список литературы по теории систем одАНЛДУ, глоссарий, примеры систем одАНЛДУ и др.

Форма ввода данных предназначена для ввода одиночных систем одАНЛДУ и их множеств, выбора решателей и генераторов, редактирования персональных настроек и управления параметрами решения и генерации.

Наиболее типичными действиями в системе Web-SynDic являются следующие.

1. Создание и завершение сессии. При первом входе в систему Web-SynDic для пользователя создается сессия, загружаются его настройки и генерируется идентификатор сессии. При последующих обращениях пользователя идентификатор сессии передается вместе с запросом. Выполняя вход в или выход из системы, пользователь закрывает предыдущую сессию и открывает новую. Для каждой сессии определено максимальное время бездействия, по истечении которого сессия автоматически завершается.
2. Запрос на генерацию и/или решение систем одАНЛДУ. Во время выполнения запроса задача помещается в очередь модулей Generator spooler или Solver spooler, а затем передается на генерацию или решение соответственно. По завершении работы генератора или решателя результаты возвращаются пользователю. В случае, если поступил запрос на генерацию и решение множества систем одАНЛДУ, результаты генерации сразу помещаются в очередь Solver spooler без дополнительного взаимодействия с пользователем. В правой части окна интерфейса пользователя регулярно обновляется и отображается текущее состояние выполнения запроса.

### Процесс разработки

В ходе разработки системы Web-SynDic использовались стандарты, технологии и средства разработки программного обеспечения [20, 21]. Проект реализован с использованием языков C++ и Java, технологии JSP на базе сервера Apache Tomcat и средств создания трансляторов JFlex и byaccj. Использование этих инструментов позволило обеспечить кросс-платформенность для сред Linux и Microsoft Windows.

Проект является командной студенческой разработкой. Средняя численность разработчиков составляет 5 человек. Заказчик проекта – Ю. А. Богоявленский (к.т.н., доцент, зав. каф. информатики и математического обеспечения Петрозаводского госуниверситета). Менеджер проекта – Д. Ж. Корзун (к.ф.-м.н., доцент). Основные разработчики – студенты К. А. Кулаков и М. А. Крышень. Эксперт со стороны заказчика и системный администратор проекта – В. А. Пономарев (ст. преподаватель). На разных этапах к проекту привлекались эксперты в области разработки программного обеспечения с кафедры информатики университета г. Хельсинки – Т. Аланко (профессор), И. Веркамо (профессор), Ю. Тайна (ст. преподаватель) и Т. Туохиниemi (преподаватель).

В качестве модели жизненного цикла разработки использовалась линейная модель с возвратами. В ходе первого этапа разработки (16.07 – 21.12.2003) была создана альфа-версия системы Web-SynDic и проведено ее массовое тестирование. В ходе закрытой экспериментальной эксплуатации (21.12.2003 – 03.08.2004) собиралась информация об ошибках и разрабатывались варианты дальнейшего развития системы. На втором этапе разработки (03.08 – 15.11.2004) в систему вносились изменения в соответствии с ранее разработанными вариантами. В результате была выпущена версия 1.0, которая подверглась повторной закрытой экспериментальной эксплуатации (15.11.2004 – 10.12.2005), по окончании которой система была опубликована для свободного доступа через Интернет.

Официальные языки проекта – русский и английский. Поддерживается полноценный набор проектной документации на основе Adaptable Process Model [21]: спецификация требований, спецификация проектирования, документация реализации и тестирования, документация пользователя, метрики проекта. В ходе разработки проекта было потрачено 2243 человеко-часа, создано 363 страницы документации, написано 11907 строк кода.

Как высокотехнологическая разработка проект Web-SynDic был удостоен первого места на межвузовском конкурсе-конференции студентов и молодых ученых Северо-Запада «Технологии Microsoft в теории и практике программирования», проходившем в г. Санкт-Петербурге в марте 2004 г. Как качественный современный информационный ресурс web-сервер проекта Web-SynDic был отмечен дипломом II степени на II конкурсе web-ресурсов ПетрГУ «В новый век с новыми технологиями» в 2005 г.

### Эксперименты в системе Web-SynDic

Эксперименты в системе Web-SynDic были разбиты на две части:

- 1) тестирование syntactic solver,
- 2) экспериментальный анализ и сравнение решателей.

В ходе реализации первой части экспериментов была произведена автоматическая генерация и решение более 1,5 миллиона систем одАНЛДУ [15]. В результате тестирования ошибок в реализации синтаксического алгоритма решения найдено не было.

В ходе проведения второй части экспериментов была построена серия уникальных систем одАНЛДУ, на основе которой сравнивались характеристики процесса решения syntactic solver и slopes. В результате было обнаружено существенное преимущество syntactic solver над slopes при увеличении размеров систем [22].

В табл. 1 в качестве примера представлена экспериментальная зависимость использования времени и памяти решателем syntactic solver от числа неизвестных в системе одАНЛДУ. Измерения выполнены для двух классов систем одАНЛДУ, полученных генераторами Gauss и Jordan соответственно. Число уравнений  $n$  варьировалось в пределах до зафиксированного числа неизвестных  $m$  (для систем одАНЛДУ всегда  $n \leq m$ ). Отметим, что подключенные на текущий момент альтернативные решатели не позволяют решать системы таких размеров.

**Таблица 1. Зависимость характеристик решения syntactic solver от числа неизвестных в системе одАНЛДУ**

| Характеристика | Метод  | Число неизвестных, $m$ |        |        |        |        |         |
|----------------|--------|------------------------|--------|--------|--------|--------|---------|
|                |        | 50                     | 100    | 200    | 300    | 500    | 1000    |
| Время, сек     | Gauss  | 0,005                  | 0,014  | 0,0369 | 0,0848 | 0,2521 | 1,5463  |
| Память, Кб     |        | 1508                   | 1756   | 2084   | 2524   | 3972   | 8168    |
| Время, сек     | Jordan | 0,0059                 | 0,0205 | 0,1123 | 0,5344 | 3,0639 | 23,5981 |
| Память, Кб     |        | 1508                   | 1756   | 2184   | 2632   | 4048   | 10188   |

Измерения выполнялись на ЭВМ с CPU Celeron 1200 МГц, RAM 512 Мб, Linux 2.6.5. Система Web-SynDic работала на платформе Java 1.4.2.08 / Apache Tomcat 5.0.19. Для каждого значения  $m$  генерировались 20 систем одАНЛДУ.

### **Заключение**

Создание высокотехнологичного ПО, использующего ресурсоемкие математические алгоритмы, требует комплексного анализа альтернативных реализаций. В случае алгоритмов решения систем одАНЛДУ система Web-SynDic предоставляет подобную вычислительную услугу, являясь виртуальной экспериментальной площадкой для проведения таких исследований.

Важным достоинством проекта является интеграция как математических проблем, так и задач, связанных с технологией разработки ПО. Это включает вычислительно сложную проблему решения НЛДУ и перспективные синтаксические алгоритмы решения систем одАНЛДУ, генерацию тестовых и эталонных систем для выполнения тестирования и экспериментального анализа решателей, технологию измерения затрат ресурсов на уровне ядра операционной системы, удаленное использование вычислительного ресурса через Интернет и кросс-платформенную разработку ПО. Как нам кажется, именно подобного рода интеграция в рамках научно-исследовательских студенческих проектов должна выступать важной компонентой подготовки высококвалифицированных специалистов международного уровня в области информатики и программирования.

### **Библиографический список**

1. Tomas A.-P, Contejean E. On Diophantine systems coming from AC-unification of higher-order patterns: exploiting symmetries. Technical Report: DCC-97-15. 1997. 8 p.
2. Pugh W., Wonnacott D. Constraint-based array dependence analysis. ACM Trans. Program. Lang. Syst. 1998. 20(3): 635–678.
3. Ghosh S., Martonosi M., Malik S. Cache Miss Equations: An Analytical Representation of Cache Misses. 11th ACM International Conference on Supercomputing. 1997, July.
4. Схрейвер А. Теория линейного и целочисленного программирования: Пер. с англ. Т. 1. М.: Мир, 1991. – 360 с.; Т. 2. М.: Мир, 1991. 342 с.
5. Корзун Д.Ж. Синтаксические алгоритмы решения неотрицательных линейных диофантовых уравнений и их приложение к моделированию структуры нагрузки канала Интернет: Дис. ... канд. физ.-мат. наук. Петрозаводск: Изд-во ПетрГУ, 2002. 185 с.
6. Кулаков К.А. Диофантова модель сети MPLS для восстановления соединений за полиномиальное время // Материалы Второй междунар. науч.-практ. конф. «Исследование, разработка и применение высоких технологий в промышленности». СПб., 2006.
7. Богоявленский Ю.А., Корзун Д.Ж. Общий вид решения системы линейных диофантовых уравнений, ассоциированной с контекстно-свободной грамматикой // Труды Петрозаводского государственного университета. Сер. «Прикладная математика и информатика». Вып. 6. Петрозаводск: Изд-во ПетрГУ, 1997. С. 79–94.
8. Корзун Д.Ж. Об одной взаимосвязи формальных грамматик и систем линейных диофантовых уравнений // Вестник молодых ученых. 2000. № 3. С. 50–56.
9. Корзун Д.Ж. Grammar-Based Algorithms for Solving Certain Classes of Non-negative Linear Diophantine Systems // Труды международного семинара «Finnish Data Processing Week at the University of Petrozavodsk

- (FDPW'2000): Advances in Methods of Modern Information Technology». Vol. 3. Петрозаводск: Изд-во ПетрГУ, 2001. С. 52–67.
10. IEEE/AIS/ACM Joint Task Force on Computing Curricula. Computing Curricula 2005. The Overview Report covering undergraduate degree programs in Computer Engineering, Computer Science, Information Systems, Information Technology, Software Engineering, September 2005 (<http://www.computer.org/curriculum> or <http://www.acm.org/education/curricula.html>).
  11. Huet G. An algorithm to generate the basis of solutions to homogenous linear diophantine equations // Information Processing Letters. 1978. Vol. 3. No. 7. P. 144–147.
  12. Filgueiras M., Tomas A.-P. Solving Linear Constraints on Finite Domains through Parsing // P. Barahona, L. Moniz Pereira, A. Porto (eds.), Proceedings of the 5th Portuguese Conference on Artificial Intelligence, Springer-Verlag, 1991. LNAI 541. P. 1–16.
  13. Domenjoud E., Tomas A.-P. From Elliot-MacMahon to an algorithm for general constraints on naturals // U. Montanari, F. Rossi (eds.), Principles and Practice of Constraint Programming (CP'95). Springer-Verlag, 1995. LNCS 976. P. 18–35.
  14. Косовский Н.К., Тишков А.В. Логика конечнзначных предикатов на основе неравенств. СПб.: Изд-во СПбГУ, 2000. 268 с.
  15. Кулаков К.А. Генерация систем однородных неотрицательных линейных диофантовых уравнений и ее приложения: Магистерская дис. Петрозаводск, 2005. 82 с. (<http://cs.karelia.ru/~kulakov/research/2005/work.pdf>)
  16. Berkelaar M. Lp\_solve algorithm. <http://www.cs.sunysb.edu/~algorithm/implement/lpsolve/implement.shtml>
  17. Makhorin A. GLPK. <http://www.gnu.org/software/glpk>
  18. Кулаков К.А. Алгоритмы генерации систем неотрицательных линейных диофантовых уравнений // Труды данной конференции.
  19. Система Web-SynDic. Web-ресурс разработки. <http://websyndic.cs.karelia.ru/site>
  20. Sommerville I. Software Engineering. 6th ed. Addison-Wesley, 2000.
  21. Roger S. Pressman. Software Engineering. A Practitioner's Approach. European adapt., 5th ed. McGraw-Hill, 2000.
  22. Кулаков К.А., Корзун Д.Ж. Generating Homogenous Systems of Equations for Testing and Experimental Analysis of Linear Diophantine Solvers // Труды международного семинара «Finnish Data Processing Week at the University of Petrozavodsk (FDPW'2003): Advances in Methods of Modern Information Technology». Vol. 5. Петрозаводск: Изд-во ПетрГУ, 2005. С. 259–278.

## ПОЛУМАРКОВСКАЯ МОДЕЛЬ ПРОЦЕССА ЗАХВАТА СРЕДЫ СТАНЦИЕЙ В БЕСПРОВОДНЫХ СЕТЯХ 802.11

**А.А. Бородин, Ю.А. Богоявленский**

Петрозаводский государственный университет

Петрозаводск, РФ

[aborod@cs.karelia.ru](mailto:aborod@cs.karelia.ru)

[ybgv@cs.karelia.ru](mailto:ybgv@cs.karelia.ru)

На сегодняшний день одной из наиболее распространенных технологий локальных беспроводных сетей является Wi-Fi, определяемая семейством стандартов 802.11. В этих сетях управление захватом передающей

среды производится всеми станциями, участвующими в передаче, на основе распределенного соревновательного метода множественного доступа CSMA/CA DCF. При росте нагрузки происходит значительное падение производительности сегмента беспроводной сети из-за роста задержек, связанных с определением станции, захватившей канал на время следующей передачи, что ведет к снижению производительности.

Для анализа производительности сегмента беспроводной сети нами разработана полумарковская модель процесса, связанного с изменением счетчика отсрочки передачи кадра и счетчика числа повторных коллизий станции, работающей в режиме насыщения, в течение соревновательного цикла. Предлагаемая модель является развитием марковской модели, предложенной Джузеппе Бьянки [1].

## 1. Модель процесса захвата среды станцией

### 1.1. Предположения модели

Ошибки, связанные с искажением кадров при передаче (например, по причине действия шума в канале связи), отсутствуют. Таким образом, кадр может быть принят с ошибкой только вследствие коллизии.

В сети отсутствуют скрытые станции и задержки распространения, сегмент беспроводной сети состоит из конечного фиксированного числа взаимодействующих станций, каждая из которых всегда имеет кадр для передачи (работает в условиях насыщения).

Основное упрощающее предположение состоит в том, что вероятность коллизии переданного кадра постоянна и не зависит от числа попыток передачи этого кадра, предпринятых ранее. Мы предполагаем известным распределение длин передаваемых кадров. В рассмотрение вводятся также длины межкадровых промежутков (определенные стандартом).

На практике хорошим приближением такой сети будут несколько станций, размещенных в одном помещении в пределах прямой видимости при отсутствии искусственных источников помех. Предположение независимости вероятности коллизии от числа попыток будет, очевидно, тем более справедливо, чем ближе находится сеть к состоянию насыщения.

### 1.2 Модель механизма захвата среды станцией

Рассмотрим  $n$  станций сегмента беспроводной сети, соревнующихся за передающую среду. В условиях насыщения MAC-протокол каждой станции после завершения каждой успешной передачи сразу же получает очередной кадр от протокола вышестоящего уровня.

Станция, захватившая канал на время следующей передачи, определяется на основе распределенного алгоритма множественного доступа с контролем несущей и предотвращением коллизий CSMA/CA (MAC-протокол семейства стандартов 802.11, реализующий алгоритм CSMA/CA, называется распределенной функцией координации, DCF). Согласно алгоритму станция связывает с принятым к передаче кадром случайное целое число в интервале  $[0, W-1]$ , называемое счетчиком отсрочки. Значение  $W$  определено стандартом и называется начальным соревновательным окном. Работа алгоритма определяется совокупностью последовательных интервалов времени, называемых временными слотами. В начале каждого слота счетчик отсрочки уменьшается на единицу, когда значение счетчика достигает нуля, станция немедленно начинает передачу кадра.

Заметим, что уменьшение времени отсрочки останавливается, как только канал опознается занятым, и, таким образом, длительность временного интервала между началами двух последовательных слотов варьируется в широких пределах от размера пустого слота до размера слота, в течение которого осуществляется передача кадра максимальной длины.

В случае, когда две или несколько станций осуществляют попытку передачи кадров в течение одного слота, происходит коллизия – искажение кадров, ведущее к неверному приему. Станция, обнаружившая коллизию, выбирает новое значение счетчика отсрочки, удваивая размер соревновательного окна.

Интервал времени от момента вовлечения кадра в соревновательный процесс до успешной передачи или исчерпания максимального числа последовательных коллизий будем называть соревновательным циклом. Таким образом, успешной передаче кадра может предшествовать один или несколько соревновательных циклов.

В связи с трудностью распознавания коллизий в беспроводной среде посредством прослушивания собственной передачи, в протоколе предусмотрена отправка специального кадра подтверждения (ACK).

Станция может осуществлять захват передающей среды, выставляя в канал биты кадра данных. Такой метод называется основным (basic access). Однако, поскольку обнаружение коллизии основывается только на получении кадра подтверждения, интервал распознавания коллизии может быть достаточно большим. Поэтому введен дополнительный метод доступа (RTS/CTS), предусматривающий предварительную отправку кадра приглашения (RTS) и получение кадра подтверждения на прием (CTS). В этом случае накладные расходы на передачу данных увеличиваются, однако скорость распознавания коллизий резко возрастает.

Следуя [1], рассмотрим  $b(t)$  – случайный процесс, представляющий значение счетчика отсрочки для заданной станции. Поскольку значение счетчика отсрочки для каждой станции зависит от истории передач, этот случайный процесс – немарковский. Обозначим  $W = CW_{\min}$ . Пусть  $m$  – максимальный уровень отсрочки, следовательно,  $CW_{\max} = 2^m W$ , и применим обозначение  $W_i = 2^i W$ , где  $i \in (0, m)$  называется уровнем отсрочки. Введем в рассмотрение также  $s(t)$  – случайный процесс, представляющий значение уровня отсрочки  $(0, \dots, m)$  станции в момент  $t$ .

Рассмотрим моменты времени  $(t_k)$ , соответствующие началам временных слотов. Тогда двумерная последовательность  $\{s_k, b_k\} = \{s(t_k), b(t_k)\}$  является марковской цепью, граф переходов которой приведен на рис. 1. В отличие от [1], нами введены в рассмотрение два дополнительных состояния «Fail» и «Success», отражающих соответственно неудачное завершение цикла по максимальному числу коллизий и передачу кадра. Заметим, что в действительности рассматривается семейство процессов, параметризованное по числу станций.

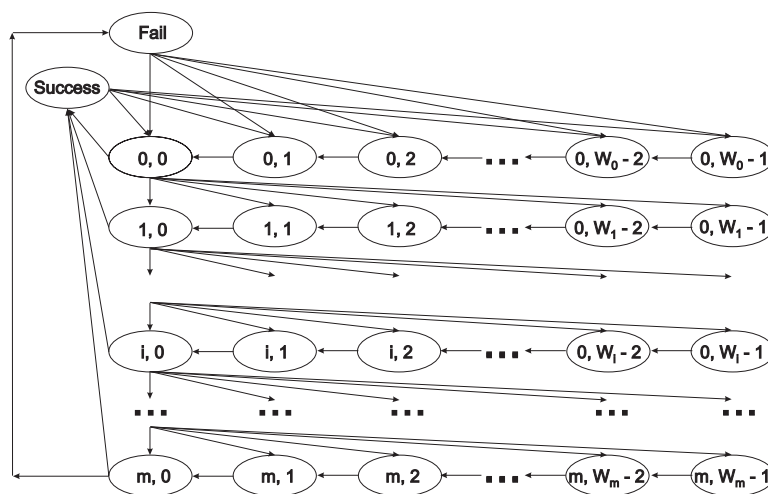


Рис. 1. Граф переходов процесса вложенной марковской цепи

Будем считать, что в начальный момент времени процесс находится в состоянии «Success». Ненулевые переходные вероятности цепи равны:

$$\left\{ \begin{array}{l} P_{\{Success|i,0\}} = 1 - p(n), i \in (0, m-1) \\ P_{\{0,k|Success\}} = \frac{1}{W}, k \in (0, W-1) \\ P_{\{Fail|m,0\}} = p(n) \\ P_{\{0,k|Fail\}} = \frac{1}{W}, k \in (0, W-1) \\ P_{\{i,k|i,k+1\}} = 1, k \in (0, 2^i W - 2), i \in (0, m) \\ P_{\{i,k|i-1,0\}} = \frac{p(n)}{2^i W}, k \in (1, 2^i W - 1), i \in (0, m) \end{array} \right.$$

### 1.3 Стационарное распределение вложенной цепи

Вложенная в процесс  $\{s(t), b(t)\}$  цепь конечна, неприводима и непериодична, следовательно, она обладает свойством эргодичности, т.е. существует стационарное распределение цепи. Аналогично [1] и [2] получим соотношения

$$\pi_{(i,k)}(n) = \frac{2^i W - k}{2^i W} p(n)^i \pi_{(0,0)}(n), i \in (0, m), k \in (0, 2^i W).$$

Используя условие нормировки, получим

$$1 = \frac{\pi_{(0,0)}(n)}{2} \sum_{i=0}^m p(n)^i (2^i W + 1).$$

Вероятность, что станция попытается передать кадр в течение произвольно выбранного слота  $\tau(n)$  и вероятность неудачной попытки  $p(n)$  могут быть выражены системой уравнений:

$$\left\{ \begin{array}{l} \tau(n) = \frac{2 \sum_{i=0}^m p(n)^i}{\sum_{i=0}^m p(n)^i (2^i W + 1)} \\ p(n) = 1 - (1 - \tau(n))^{n-1} \end{array} \right.$$

Эта система может быть решена численно.

### 1.4 Стационарное распределение полумарковского процесса

Двумерный процесс  $\{s(t), b(t)\}$ , является полумарковским процессом, управляемым вложенной цепью  $\{s_k, b_k\}$ . Введение полумарковского процесса позволяет учесть в модели непостоянство длин временных слотов, возникающее из-за остановки уменьшения счетчика отсрочки при опознании канала как занятого. Множество всех состояний процесса  $\{s(t), b(t)\}$  обозначим  $S$ .

Предполагая известным распределение интервалов времени между переходами, можно определить стационарные вероятности полумарковского процесса, определенного на переходах вложенной цепи.

Пусть  $E_s$  — средняя длительность слота успешной передачи,  $E_c$  — средняя длительность коллизийно-го слота,  $E_i$  — длительность пустого слота. Значение  $E_i = \sigma$  зафиксировано стандартом и зависит от кон-

кретной технологии, используемой на физическом уровне. Стандарт определяет также длительность передачи заголовка кадра данных H, кадров RTS, CTS, ACK, длины межкадровых промежутков SIFS, DIFS, EIFS.

Пусть  $v_l$  — вероятность появления кадра с полем данных, длительность передачи которого равна  $l$ . Тогда средняя длительность слота успешной передачи для метода передачи «basic access» определяется как

$$E_s = H + \sum_{l=l_{\min}}^{l_{\max}} l v_l + SIFS + \sigma + ACK + DIFS + \sigma,$$

а для метода «RTS/CTS»

$$E_s = RTS + 3SIFS + 4\sigma + \sum_{l=l_{\min}}^{l_{\max}} l v_l + SIFS + \sigma + ACK + DIFS + CTS + H.$$

Длительность коллизийного слота для метода «basic access» вычисляется как

$$E_c = H + \sum_{l=l_{\min}}^{l_{\max}} l v_l + DIFS + \sigma,$$

а для метода «RTS/CTS»

$$E_c = RTS + DIFS + \sigma.$$

Пусть  $E_b$  — средняя длительность отложенного слота, в течение которого станция не предпринимает попыток передачи, поскольку связанный с ней счетчик отсрочки не равен нулю. Отложенный слот может быть пустым, если ни одна из оставшихся станций сегмента не передает, успешным, если передает только одна, и коллизийным, если станций, осуществляющих передачу в течение слота, несколько. Поэтому имеем

$$E_b(n) = (1 - \tau(n))^n E_i + n \tau(n) (1 - \tau(n))^{n-1} E_s + (1 - (1 - \tau(n))^n - n \tau(n) (1 - \tau(n))^{n-1}) E_c$$

Пусть  $E_t(n)$  — средняя длительность слота передачи. С вероятностью  $p(n)$  попытка передачи испытывает коллизию, в противном случае передача осуществляется успешно, т. е.

$$E_t(n) = (1 - p(n)) E_s + p(n) E_c.$$

Зная долю времени, в течение которого система пребывает в состоянии  $(i, k)$ , при условии, что следующий переход будет произведен в состояние  $(j, h)$ , мы можем получить долю времени, в течение которого станция осуществляет передачу

$$T_i(n) = \frac{\tau(n) E_t(n)}{(1 - \tau(n)) E_b(n) + \tau(n) E_t(n)}.$$

Анализ последнего уравнения позволяет сделать вывод, что фактическая вероятность передачи при небольшом количестве станций в сегменте беспроводной сети существенно больше, чем это получено в [1].

## 2. Характеристики производительности сегмента беспроводной сети

На основе стационарных вероятностей полумарковского процесса получим выражения для основных характеристик производительности метода множественного доступа, а именно для пропускной способности в состоянии насыщения и средней задержки обработки кадра, вовлеченного в соревновательный цикл.

### 2.1. Пропускная способность в состоянии насыщения

Нормализованной пропускной способностью в состоянии насыщения  $T(n)$  назовем долю времени, в течение которой канал используется для успешной передачи полезных данных. (Заметим, что в данном случае под полезными данными будут пониматься сообщения вышележащих уровней, в том числе заголовки.)

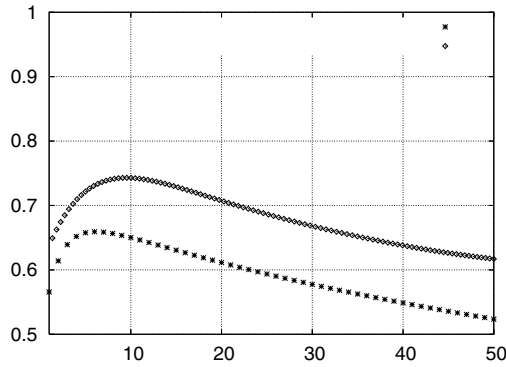


Рис. 2. Пропускная способность насыщения для метода «basic access»

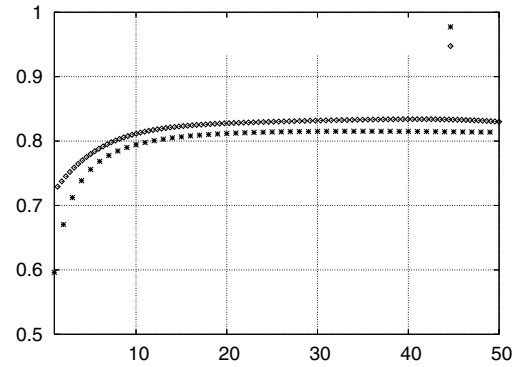


Рис. 3. Пропускная способность насыщения для метода «RTS/CTS»

Пусть  $E_p$  — средняя длина поля данных кадра. Тогда  $T(n)$  может быть выражена как

$$T(n) = \frac{n\tau(n)(1-p(n))E_p}{(1-\tau(n))E_b(n) + \tau(n)E_t(n)}.$$

Для того чтобы сравнить результаты численных расчетов на основе полумарковской модели с полученными в [1], мы используем те же системные параметры и предполагаем одинаковый размер всех кадров (длина поля данных 8184 бита).

Рис. 2 и 3 демонстрируют нормализованную пропускную способность насыщения, достижимую протоколом DCF, соответственно при использовании метода доступа «basic access» и «RTS/CTS». Заметим, что в случае метода «basic access» разница между результатами анализа марковской и полумарковской моделей составляет около 10%.

## 2.2. Средняя задержка обработки кадра станцией

В отличие от [1], наша модель позволяет легко получить важную с практической точки зрения оценку производительности сегмента беспроводной сети — задержку обработки кадра станцией.

Получим среднюю длительность интервала времени, предшествующего успешной передаче кадра. При этом будем предполагать, что кадр, израсходовавший все попытки передачи и перешедший в состояние «Fail», немедленно переходит на новый соревновательный цикл.

Пусть  $E_{\text{cycle}}(n)$  — средняя длина соревновательного цикла, а  $E_{\text{ncycles}}(n)$  — среднее число циклов, предшествующих передаче. Тогда средняя задержка кадра может быть вычислена как

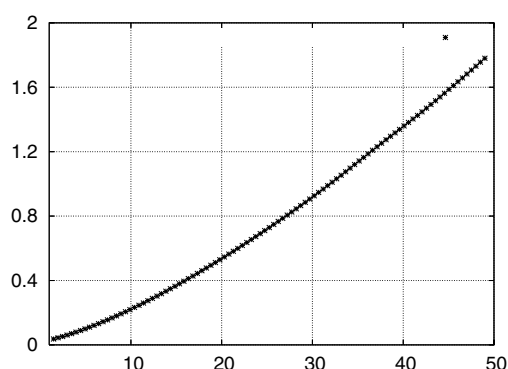
$$E_{\text{delay}}(n) = E_{\text{cycle}}(n) \times E_{\text{ncycles}}(n).$$

$E_{\text{cycle}}(n)$  может быть вычислена как среднее время возвращения в объединенное состояние «Fail-Success». В силу структуры полумарковского процесса  $E_{\text{cycle}}(n)$  может быть получено как

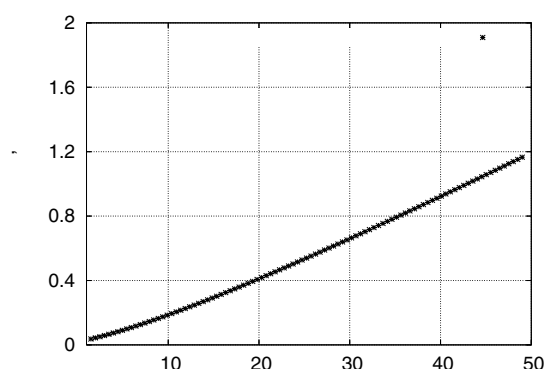
$$E_{\text{cycle}}(n) = (1-p(n)^{m-1})E_s + \sum_{r=1}^m \left( \frac{1}{2^r W} E_c + \frac{2^r W - 1}{2} E_i \right) p(n)^r (1-p(n)^{m-r}).$$

Получим  $E_{\text{ncycles}}(n)$ :

$$E_{\text{ncycles}}(n) = \left( (1-p(n)^m) + m \frac{p(n)^{2m-1}}{1-p(n)^m} \right).$$



**Рис. 4. Задержка обработки в режиме насыщения для метода «basic access»**



**Рис. 5. Задержка обработки в режиме насыщения для метода «RTS/CTS»**

На рис. 4 и 5 проиллюстрирована зависимость величины задержки обработки от числа станций в сегменте в случае, когда сегмент беспроводной сети работает в режиме насыщения (при тех же системных параметрах, что и в примере для пропускной способности).

### **Заключение**

В работе рассмотрен процесс захвата среды станцией в сегменте беспроводной сети 802.11. Построена и исследована модель процесса захвата, представляющая двумерный полумарковский процесс. Для построенного процесса получено стационарное распределение вероятностей, на основе которого получены выражения для характеристик производительности сегмента беспроводной сети: нормализованной пропускной способности и задержки обработки кадра станцией в соревновательном процессе. Использование полумарковского процесса позволило существенно уточнить модель [1].

### **Библиографический список**

1. Bianchi G. Performance Analysis of the IEEE 802.11 Distributed Coordination Function // IEEE Journal on Selected Areas in Communication. March 2000. Vol. 18. No. 3. P. 535–547.
2. Wu H., Peng Y., Long K., Cheng S., Ma J. Performance of Reliable Transport Protocol over IEEE Wireless LAN: Analysis and Enhancement // Proc. IEEE ICC 2002. P. 599–607.

### **Semi-Markov Model of the 802.11 Distributed Coordination Function**

Recent investigations have shown that the performance of a Wi-Fi segment significantly depends on its workload and decreases very fast with the growth of the competing stations number. This misbehavior is mainly connected with the distributed coordination function (DCF), which is the 802.11 contention resolution method. Several improvements of DCF have been proposed by researchers but the most of them are based on simulations rather than on analytical models.

In our work we present the semi-Markov model of the contention process of a wireless station. We revise the previous investigations results of the generic DCF algorithm and its enhancements EIED, SD, PDCF and others from our model point of view. The comparison of the normalized throughput and average delay estimations was carried out for the DCF and its modifications. The theoretical results are illustrated by numerical examples.

## ИНФОРМАЦИОННЫЕ ОСНОВЫ И ПРИНЦИПЫ ПОСТРОЕНИЯ ИНТЕЛЛЕКТУАЛЬНЫХ ВИДЕОСИСТЕМ

**В.П. Бююн**

Институт кибернетики им.В.М. Глушкова НАН Украины

Киев, Украина

*boyun@svitonline.com*

Рассмотрены особенности и основы цифрового представления изображений и видеопоследовательностей в системах реального времени. Сформулированы требования к видеосенсорам для реализации рассмотренных способов цифрового представления изображений и принципы построения интеллектуальных видеосистем. Описана разработанная интеллектуальная видеокамера с программируемыми параметрами считывания и предварительной обработкой изображений.

## THE DEVELOPMENT OF COMPUTER SCIENCE IN BULGARIA AND THE COOPERATION OF THE EASTERN EUROPEAN COUNTRIES IN THE CREATION OF COMPUTER SERIES

**K.L. Bojanov**

Institute of Parallel Processing of Information BAN

Sofia, Bulgaria

*bojanov@acad.bg*

The report gives information on the design and production of the first computers and electronic calculators in Bulgaria.

The emphasis is on the cooperation with the USSR and Eastern European countries (Hungary, Czechoslovakia, Poland, GDR, Romania, etc.) during the design, creation and production of computer systems and computer devices. The main trends of the research and production activities for mainframes, mini computers and personal computers are described, so that the produced systems are compatible with similar systems, manufactured in the USA and other Western countries.

The most important directions and ways of cooperation, including information exchange and inter country coordination are presented. Another topic is the way of cooperation between the research institutes and scientific academies of the countries, members of the former COMECON.

Special attention is given to the economic mechanisms and approaches for system assembly and the adoption and of legislation documents in the countries. The links between the various scientific and production organizations is shown with the role of their leadership.

Data on the nomenclature of designed products, produced elements, service and system supplies in various countries, as well as the cooperation between scientific and industrial organizations with the Academies of Sciences is presented.

## ТРОИЧНАЯ ДИАЛЕКТИЧЕСКАЯ ИНФОРМАТИКА

Н.П. Брусенцов, Ю.В. Владимирова

Московский государственный университет

Москва, РФ

*ramil@cs.msu.ru*

Неполноценность современной двоичной информатики, наглядно проявляющаяся в невозможности естественного представления чисел со знаком двоичным кодом, обусловлена неадекватностью ее основы – двухзначной логики. Эта так называемая «классическая» формальная логика, базирующаяся на априорном «законе исключенного третьего», не составляет совершенного отображения реальности, не согласуется со здравым смыслом. Потому она, в отличие, например, от арифметики, практически не используется для решения реальных проблем. По существу это логика искусственного дискретного мира двоичных компьютеров. В ней отсутствуют модальности, возможность неотличима от необходимости. Даже фундаментальнейшее логическое отношение содержательного следования вырождено в «материальную импликацию», парадоксы которой выдающиеся логики упорно, но тщетно пытаются преодолеть на протяжении почти ста лет.

Понятно, почему не удастся сделать логику школьным предметом, несмотря на то, что воспитание логичного мышления – первейшая задача школы, как и вузов, в которых логика, хотя и преподается, но не способствует совершенствованию умов. Положение крайне ухудшилось с компьютеризацией образования: ведь при практической безрезультатности разговоров об интеллектуализации машинной обработки информации и об информационной безопасности совершается повсеместное подавление творческого человеческого интеллекта ходульной двухзначной логикой.

Основоположником логики справедливо считается Аристотель, создавший систему доказательного умозаключения – силлогистику, которая все еще остается непревзойденным интеллектуальным инструментом. Силлогистика диалектична, и в ней нет парадоксов, но она не отобразима в современных логических исчислениях, что стало поводом для подозрений, будто у Аристотеля что-то не так, например, будто он не признавал пустых множеств. На самом деле в силлогистике налицо не только пустые, но и нечеткие множества, открытые Л. Заде в 1965 г. и все еще «не освоенные» современной логикой. Но принципиальное отличие логики Аристотеля от современной «классической» в том, что она не двухзначна, а трехзначна [3]. Вопреки «закону исключенного третьего» у Аристотеля наряду с «необходимо есть» и «необходимо нет», имеется третье – «возможно, есть и, возможно, нет». Трехзначность присуща отношению следования, исчерпывающе определенному Аристотелем в «Первой аналитике»:

«...Когда два [объекта] относятся друг к другу так, что если есть один, необходимо есть и второй, тогда, если нет второго, не будет и первого, однако если второй есть, то не необходимо, чтобы был первый. Но невозможно, чтобы одно и то же было необходимо и когда другое есть и когда его нет».

В силлогистике отношение следования представлено общеутвердительной посылкой «Всякое  $x$  есть  $y$ », сущность которой в том, что всякая  $x$ -вещь необходимо есть  $xy$ -вещь, а всякая  $y'$ -вещь (не- $y$ -вещь) необходимо есть  $x'y'$ -вещь. При этом естественно исключены (невозможны)  $xy'$ -вещи, поскольку  $x$  необходимо должно быть  $y$  (не может быть  $y'$ ). Однако вместе с тем  $x'y$ -вещи не исключены, они **ВОЗМОЖНЫ**, но не необходимы в силу наличия  $xy$ - и  $x'y'$ -вещей. Если же наряду с «Всякое  $x$  есть  $y$ » выполняется «Всякое  $y$  есть  $x$ », то будут исключены как  $xy'$ -, так и  $x'y$ -вещи, т.е. получится отношение эквивалентности – « $x$  равнозначно  $y$ », выражимое в двухзначной логике: «равнозначно / неравнозначно». Для отношения же следования требуются три значения: «необходимо есть» / «возможно, но не необходимо» / «невозможно», поэтому двухзначной импликации для адекватного выражения следования недостаточно. При наличии  $xy'$ -вещей

следование невозможно, а если  $xy'$ -вещей нет, то следование не исключено, т.е. возможно, но не необходимо. В случаях же несуществования  $x$ -вещей, либо  $y'$ -вещей, импликация вообще не выражает никакой взаимосвязи между терминами, не составляет двухместного отношения. В этих случаях другой термин волен принимать любое значение независимо.

Для устранения «парадоксов» импликации достаточно предотвратить эти случаи неперенности (константности) ее терминов. Так, строгая импликация Льюиса, представляющая собой несуществование  $xy'$ -вещей, т.е.  $V'xy'$ , парадоксально выполняется при несуществовании  $x$ -вещей, а также при несуществовании  $y'$ -вещей, т.е. при  $V'x$  и при  $V'y'$ . Парадоксов не станет, если наряду с  $V'xy'$  потребовать  $Vx$  и  $Vy'$  – существования  $x$ -вещей и  $y'$ -вещей. В результате импликация Льюиса превращается в аристотелево необходимое следование  $VxV'xy/Vy'$  [1]. В совершенной дизъюнктивной нормальной форме этого отношения –  $VxyV'xy/Vx'y'$  – очевидна его трехзначность: наряду с принадлежностью рассматриваемому подмножеству декартова произведения  $\{x, x'\} \times \{y, y'\}$  членов  $xy$ ,  $x'y'$  и антипринадлежностью  $xy'$  умалчивается член  $x'y$ . Умалчиванием выражена третья разновидность принадлежности – привходящая, возможная, но не необходимая. Подмножество, допускающее привходящую принадлежность, нечетко, а представленное им отношение трехзначно.

В математической логике отклонение от аристотелева истолкования общеутвердительной посылки «Все  $A$  суть  $B$ », превратившее ее из содержательного следования в двухзначную импликацию, Гильберт и Аккерман оправдывали потребностями математических применений логики, «где класть в основу аристотелево понимание было бы нецелесообразно» [2, с. 79]. Они не придавали значения тому, что логика при этом утратила содержательность, полагая, что их логическое исчисление «делает возможным успешный охват проблем, перед которыми принципиально бессильно чисто содержательное логическое мышление» [2, с. 17].

На самом деле бессодержательной логику сделали 2,3 тыс. лет тому назад античные стоики, также стремившиеся к запредельной абстрактности, которая была осуществлена посредством «высказываний», подчиненных «закону исключенного третьего», допускавшему для высказывания только два значения «истинности» – «истина» / «ложь». Именно двухзначностью адекватная живой реальности силлогистика Аристотеля превращена в мертвую схоластику. Математическая логика лишь выразила эту «классику» в строгих алгебраических формах, с очевидностью выявивших ее неадекватность [3].

Стоики «скомпенсировали» отсутствие в их логике отношения следования с необходимостью осуществлением умозаключений по правилам *modus ponens* и *modus tollens*. В математической логике также указывают на то, что импликация – это не следование: «Соотношение "если  $X$ , то  $Y$ " не следует понимать как выражение для отношения основания и следствия. Напротив, высказывание  $X \rightarrow Y$  истинно всегда уже в том случае, когда  $X$  есть ложное или же  $Y$  истинное высказывание» [2, с. 20]. Однако вместе с тем в математической логике даже сами ее основоположники отождествляют двухзначную импликацию с трехзначным аристотелевым следованием, поскольку и то и другое ассоциируется с суждением «Все  $A$  суть  $B$ ». В результате с точки зрения математической логики признаны ошибочными безупречные в действительности модусы силлогистики *darapti*, *bamalip*, *felapton*, *fesapo* [2, с. 79] и отвергается силлогизм подчинения частного общему, ибо согласно отклонению от аристотелева истолкования в суждении «Все  $A$  суть  $B$ » не содержится с необходимостью «Некоторые  $A$  суть  $B$ ».

Ян Лукасевич, изобретая в 1920 г. трехзначную модальную логику, в своей обстоятельной книге «Аристотелевская силлогистика с точки зрения современной формальной логики» [4] алгебраически «доказал» путем отождествления трехзначного следования с материальной импликацией, что цитированное выше аристотелево утверждение: «Но невозможно, чтобы одно и то же было необходимо и когда другое есть, и

когда его нет» неверно. Оно неверно с точки зрения логики, не соблюдающей основной логический закон – закон тождества. Суждение «Все  $A$  суть  $B$ », выражающее отношение следования  $B$  из  $A$ , нельзя отождествлять с «Ни одно  $A$  не есть не- $B$ », которым в естественном языке представлено двухзначное отношение импликации.

Беда, по-видимому, в том, что логика немислима без отношения следования, которое в сущности трехзначно и в двухзначной логике естественно отсутствует. Отношение, называемое импликацией, подобно следованию, выражается тем же «если... то...» и обозначается той же стрелкой  $\rightarrow$ . Не удивительно, что импликацию принимают за следование. Но если без следования нет логики, то логика с импликацией в качестве следования тем более не логика. В ней из несуществующего следует «все, что угодно», из « $2 \times 2 = 4$ » следует, что «снег бел».

Это все та же ущербная, пренебрегающая здравым смыслом двухзначная логика, и результат ее внедрения, естественно, не оправдывает ожиданий. Так, в книге Т. Оппенгеймера [5] неопровержимо доказана пагубность компьютеризации обучения в школах США. Автор настаивает на удалении компьютеров из школы, что едва ли осуществимо в сложившейся ситуации. Но ведь корень зла не в компьютерах, а в заложенной в них примитивной, противоестественной логике дискретного двоичного мира, которая прививается учащимся, блокируя их способность овладеть логикой мира реального. Окажись в компьютерах благоразумная естественная логика, результат компьютеризации обучения был бы диаметрально противоположным.

Однако адекватной (диалектической) логики сегодня нет не только в компьютерах, но и во всей «науке о мышлении», даже в той ее части, которая не подчинена догматическому «закону исключенного третьего» (неправомерно приписываемому Аристотелю) и сосредоточилась на изобретении недвухзначных («неаристотелевых») логик. Изобретательство это безуспешно, потому что носит чисто формальный характер. Если бы исследовали проблему по существу, то обнаружили бы, во-первых, что логика Аристотеля трехзначна, а во-вторых, что трехзначность – необходимое, но не достаточное условие адекватности логики. Логика же Аристотеля адекватна, поэтому изобретать неаристотелевы логики нет смысла.

Отрадным исключением является «Символическая логика» Льюиса Кэрролла [6], не нашедшая (как и аристотелева) должного понимания и надлежащего развития. У Кэрролла нет ни бессодержательных «истинных»/ «ложных» высказываний, ни «закона исключенного третьего». Его логика исследует суждения, выражающие взаимосвязи (отношения) вещей, охарактеризованных совокупностями признаков (особенностей). «Во Вселенной множество *вещей*... Вещи обладают *признаками*... Любой признак или любую совокупность признаков будем называть также *особенностью* вещи».

Суждение рассматривается как естественноречевое выражение отношения, которым связаны обозначаемые посредством *терминов*  $x, y, z, \dots$  особенности вещей. Вместе с тем сущность того же отношения Кэрролл наглядно отображает на изобретенной им диаграмме и алгебраически так называемым «методом индексов», позволяющим формально получить содержательное умозаключение из данных суждений, если оно существует.

Диаграмма Кэрролла внешне тождественна таблице истинности Пирса, используемой для определения булевых функций. Однако истолковывается представленное диаграммой не экстенционально (не как класс вещей), а интенционально – как множество вещей, а точнее, как подмножество декартова произведения попарно противоположных особенностей. Кроме того, клеткам диаграммы присваивается не одно из двух, а одно из трех значений – наряду с содержащими «0» либо «1» допускаются незанятые (пустые) клетки, означающие несущественность принадлежности соответствующих им вещей представленному на диаграмме подмножеству. Правда, сам Кэрролл понимал значение «1» как существование, значение «0» как несуществование вещи, а незанятость клетки у него не утверждает ни того, ни другого.

Например, отношение строгой импликации Льюиса  $V'xy'$  на двухтерминной диаграмме Кэрролла представлено единственным значением «0» в  $xy'$ -клетке. Кэрролл выражал это отношение тройко: общеотрицательным суждением «Ни один  $xy'$  не существует», либо «Ни один  $x$  не есть  $y'$ », либо «Ни один  $y'$  не есть  $x$ ».

Общеутвердительное суждение «Все  $x$  суть  $y$ » у Кэрролла содержало в себе необходимо следующее из него частноутвердительное «Некоторые  $x$  суть  $y$ », равносильное суждению существования «Некоторые  $xy$  существуют». Кэрролл называл «Все  $x$  суть  $y$ » двойным суждением, эквивалентным двум суждениям: «Ни один  $x$  не есть  $y'$ » и «Некоторые  $x$  суть  $y$ », т.е.  $V'xy/Vxy$  – на диаграмме «0» в клетке  $xy'$  и «1» в клетке  $xy$ .

Ясно, что это уже не импликация (один из ее парадоксов устранен), но еще и не полноценное аристотелево следование. Не учтена контрапозитивность следования – оплошность, присущая, по-видимому, всем известным попыткам алгебраизации силлогистики. Восполнением этого упущения [7] «Символическая логика» Кэрролла превращается в наредкость стройное и безупречное изложение категорической силлогистики Аристотеля – фундамента диалектической логики.

Важнейшим критерием содержательности логики оказался выявленный в основании силлогистики Аристотеля диалектический *принцип сосуществования противоположностей* [8]. Контрапозитивность общеутвердительного суждения, как и симметричность отношения, выражаемого общеотрицательным суждением, представляют собой очевидные проявления этого принципа. Сущность же его в том, что обозначаемые посредством терминов  $x, y, z, \dots$  первичные (несоставные) особенности  $x, x', y, y', z, z', \dots$  обретают смысл только в результате сопоставления вещей, обладающих противоположными особенностями, например  $x$ -вещи с  $x'$ -вещью.

Другими словами, принцип сосуществования противоположностей означает, что подмножеству декартова произведения  $\{x, x'\} \times \{y, y'\}$ , отображающему содержательное отношение, необходимо принадлежат (сосуществуют в нем) все попарно противоположные особенности –  $VxVx'VyVy'$ . На диаграмме Кэрролла выражение  $VxVx'VyVy'$  отображается наличием фишки существования «1» на каждой из четырех внутренних стенок, означаящим непустоту классов  $x, x', y, y'$ .

Это адекватный реальности Универсум Аристотеля (УА) – основа содержательной логики [8]. Именно в нем льюисова импликация  $V'xy'$  и кэрроллово  $VxV'xy'$  необходимо превращаются в полноценное следование:

$$(V'xy')(VxVx'VyVy') \equiv VxyV'xy/Vx'y';$$

$$(VxV'xy')(VxVx'VyVy') \equiv VxyV'xy/Vx'y'.$$

Несуществование какой-либо из возможных на двухтерминной диаграмме вещей, например  $xy$ -вещи, в УА означает существование двух смежных с ней вещей:

$$(V'xy)(VxVx'VyVy') \equiv V'xyVxVy \equiv V'xyVxy/Vx'y'.$$

Существование же, например,  $xy'$ -вещи в силу принципа сосуществования противоположностей необходимо влечет также существование ее антипода –  $x'y$ -вещи. Поэтому частноутвердительная и частноотрицательная посылки силлогистики оказываются двойными и их всего две, а не четыре:

$$Ixy \equiv AxyAyx \equiv VxyVx'y';$$

$$Oxy \equiv ExyEx'y' \equiv Vxy/Vxy'.$$

Вместе с тем общих посылок оказывается не две, а четыре, впрочем, сводящиеся к одной инвертированием терминов:  $Exy \equiv Axy', Ex'y' \equiv Ax'y, Ayx \equiv Ax'y'$ .

Алгебра силлогистики, соответствующая кэрроллову истолкованию его диаграммы с фишками, символизирующими существование и несуществование вещей, сопоставленных клеткам и разделяющим клетки «стенкам», аналогична «методу индексов», но вместо индексов использует префиксный функтор существования  $V$  – «дизъюнкт» (интегральная дизъюнкция, подобная интегральной сумме  $\Sigma$ ) и его инверсию  $V'$  –

символ несуществования. Представленное на диаграмме отношение отображается конъюнкцией дизъюнктов, неинвертированных и инвертированных, а члены конъюнкции, соответствующие пустым клеткам, умалчиваются.

Дизъюнктивная алгебра компьютеризована путем отображения диаграммы-матрицы в одномерную ДК-шкалу тритов [9], сопоставленных клеткам диаграммы в порядке «убывания» соответствующих им особенностей:  $xy$ ,  $xy'$ ,  $x'y$ ,  $x'y'$ . Каждый трит принимает одно из трех значений: «+» – существование, «–» – несуществование, «0» – умалчивание. Например, следование  $(x \Rightarrow y) \equiv \forall xy \forall' xy' \forall x' y'$  отображается значением четырехтритной шкалы:  $+ - 0 +$ , частноотрицательная посылка  $Oxy$  кодируется значением:  $0 + + 0$ .

Всего в силлогистике восемь двухместных отношений [10]:

$$\begin{aligned} Axy &\equiv Ay'x' \equiv Exy' \equiv Ey'x \equiv + - 0 + \\ Ayx &\equiv Ax'y' \equiv Eyx' \equiv Ex'y \equiv + 0 - + \\ Exy &\equiv Eyx \equiv Axy' \equiv Ayx' \equiv - + + 0 \\ Ex'y' &\equiv Ey'x' \equiv Ax'y \equiv Ay'x \equiv 0 + + - \\ Ixy &\equiv Ix'y' \equiv Iyx \equiv Iy'x' \equiv Oxy' \equiv Oyx' \equiv Ox'y \equiv Oy'x \equiv + 0 0 + \\ Oxy &\equiv Oy'x' \equiv Oyx \equiv Ox'y' \equiv Ixy' \equiv Iy'x \equiv Ix'y \equiv Iyx' \equiv 0 + + 0 \\ x \Leftrightarrow y &\equiv AxyAyx \equiv Exy'Eyx' \equiv + - - + \\ x \Leftrightarrow y' &\equiv ExyEyx \equiv Axy'Ayx' \equiv - + + - \end{aligned}$$

Компьютеризованное доказательство умозаключений (правильных модусов силлогизма) осуществляется путем представления двухтерминных посылок трехтерминными шкалами, из пересечения которых элиминацией среднего термина выявляется искомое заключение, если оно существует. Например, модус *barbara*:  $AyzAxu \Rightarrow Axz$  в трехтерминных  $x, y, z$ -шкалах реализуется так:

$$\begin{aligned} Ayz &\equiv + - 0 + + - 0 + \\ Axy &\equiv + + - - 0 0 + + \\ Ayz \cap Axy &\equiv + - - - 0 0 0 + \end{aligned}$$

Элиминация  $y$  дает  $x, z$ -шкалу  $+ - 0 +$ , т.е.  $Axz$ .

Подчиненность частных посылок общим доказывается пересечением кодирующих эти посылки шкал. Так, подчинение  $Axy \Rightarrow Ixy$ , равносильное  $AxyIx'y' = Axy$ , удостоверяется пересечением  $+ - 0 + \cap + 0 0 + = + - 0 +$ .

В базируемой на сосуществовании противоположностей силлогистике доказуемы все сомнительные с точки зрения классической логики модусы, а также ряд модусов, упущенных традиционной силлогистикой. Например, из посылок сомнительного модуса *bamalip* на самом деле следует не только частное, но и общее заключение:

$$\begin{aligned} Azu &\equiv + 0 - + + 0 - + \\ Ayx &\equiv + + 0 0 - - + + \\ Azu \cap Ayx &\equiv + 0 - 0 - - + + \end{aligned}$$

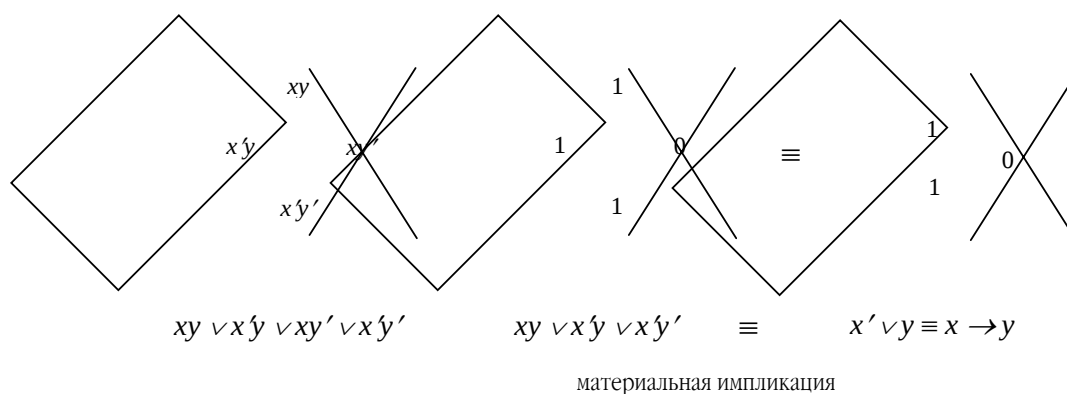
Элиминировав  $y$ , имеем  $+ 0 - + \equiv Azx \equiv Ax'z'$ , т.е.  $AzuAyx \Rightarrow Azx$ .

Коррекцией традиционной теории является доказательство отрицаемого ею модуса первой фигуры  $IyzAxu \Rightarrow Ixz$ :

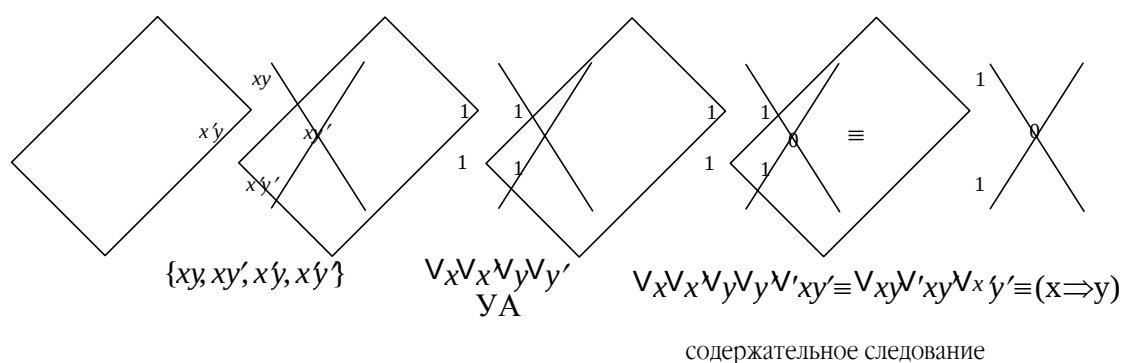
$$\begin{aligned} Iyz &\equiv + 0 0 + + 0 0 + \\ Axy &\equiv + + - - 0 0 + + \\ Iyz \cap Axy &\equiv + 0 - - 0 0 0 +, \end{aligned}$$

что по исключении  $y$  есть  $+ 0 0 +$ , т.е.  $Ixz$ .

Таким же образом доказуема правильность другого непризнанного модуса первой фигуры  $\Gamma yzExy \Rightarrow Oxz$ , а также аналогичных модусов других фигур.



**Экстенциональная интерпретация (таблица Пирса, булева алгебра классов)**



**Интенциональная интерпретация (диаграмма Кэрролла в Универсуме Аристотеля UA)**

### Библиографический список

1. Брусенцов Н.П. Трехзначная интерпретация силлогистики Аристотеля // Историко-математические исследования. Вторая серия. Вып. 8 (43). М.: «Янус-К», 2003.
2. Гильберт Д., Аккерман В. Основы теоретической логики. М.: ИЛ, 1947.
3. Лосев А.Ф. Критические заметки о буржуазной математической логике // Историко-математические исследования.
4. Лукасевич Я. Аристотелевская силлогистика с точки зрения современной формальной логики. М.: ИЛ, 1959.
5. Oppenheimer T. The Flickering Mind: The False Promise of Technology in the Classroom and How Learning Can Be Saved. – New York, Random House, 2003.
6. Кэрролл Л. Символическая логика // Кэрролл Л. История с узелками. М.: «Мир», 1973.
7. Брусенцов Н.П. Диаграммы Льюиса Кэрролла и аристотелева силлогистика // Вычислительная техника и вопросы кибернетики. Вып. 13. М.: Изд-во Моск. ун-та, 1977.
8. Брусенцов Н.П. Блуждание в трех соснах: (Приключения диалектики в информатике). М.: SVR-Аргус, 2000. (<http://www.computer-museum.ru/books/archiv/3pines.zip>)

9. Брусенцов Н.П., Владимирова Ю.С. Троичная компьютеризация логики // Математические методы распознавания образов: 12-я Всероссийская конференция: сб. докл. М.: МАКС Пресс, 2005.
10. Брусенцов Н.П. Реанимация аристотелевой силлогистики // Реставрация логики. М.: Фонд «Новое тысячелетие», 2005.

## СМЕШАННЫЕ ВЫЧИСЛЕНИЯ В НОВОСИБИРСКЕ

**М.А. Бульонков**

Институт систем информатики им. А.П. Ершова СО РАН  
Новосибирск, РФ  
*mike@iis.nsk.su*

В этих заметках я хочу рассказать о том, как развивались смешанные вычисления в Новосибирском Вычислительном центре и позднее в ИСИ СО РАН. Вполне возможно, я не смогу упомянуть обо всех событиях и людях, которые были причастны к работам в этой области, но ни в коем случае не с целью уменьшить их значимость.

### Андрей Петрович Ершов – проекции Футамуры

К идее смешанных вычислений А.П. Ершов пришел на основе работ в области разработки трансляторов, оптимизации и преобразования программ. В первых работах формальное определение имело следующий вид: смешанный вычислитель определялся как программный процессор, который получает на вход некоторое представление программы и часть входных данных и получает на выходе преобразованную программу, часть результата и данные, которые требуют дополнительной обработки.

Далее Андрей Петрович заметил, что поведение транслятора очень напоминает поведение смешанного вычислителя: точно так же действия по анализу транслируемой программы чередуются с генерацией объектного кода. Кульминацией этих рассуждений стало открытие соотношений, которые связывают интерпретацию программы и ее трансляцию в объектный код. Для этого смешанный вычислитель трактовался в более узком смысле: без промежуточных данных и частичного результата:

$$\text{Mix}(p, x) = p_x$$

так, что  $p(x, y) = p_x(y)$ .

Назовем последнее равенство *уравнением смешанного вычислителя*. Пусть далее имеется интерпретатор *int* некоторого языка *L*, записанный в языке, программы которого может обрабатывать *mix*. Тогда

$$\text{Int}(p, d) = p(d),$$

для любой программы *p* в языке *L* и ее данных *d*. Тогда, подставляя *int* в уравнение смешанного вычислителя, получаем

$$\text{int}(p, d) = \text{int}_p(d),$$

откуда  $\text{int}_p(d) = p(d)$ .

То есть программы  $\text{int}_p$  и *p* эквивалентны, но заметим, что записаны они в разных языках: *p* – в языке *L*, а  $\text{int}_p$  – в языке интерпретатора. Иными словами, смешанные вычисления в применении к интерпретатору реализуют трансляцию, а  $\text{int}_p$  является объектным кодом *p*.

Продолжая такие игры с формулами, наиболее интересные соотношения можно получить, если предположить, что смешанный вычислитель обладает свойством *самоприменимости*. Для этого он, по крайней мере, должен быть записан в том же языке, для которого и предназначен. Оставляем читателю самому доказать, что

$$mix_{int}(p) = int_p \text{ и } mix_{mix}(int) = mix_{int}.$$

Таким образом,  $mix_{int}$  преобразует программу в объектный код и, следовательно, реализует функцию транслятора, а  $mix_{mix}$  преобразует семантику языка, заданную в виде интерпретатора, в транслятор и, следовательно, является генератором трансляторов.

Всякий, кто имел хоть приблизительное представление о трансляторах и интерпретаторах и узнавал эти соотношения, вначале не верил, что такое возможно, и искал какой-то подвох, а затем приходил в восхищение от красоты, ясности и глубины заложенного в них смысла<sup>1</sup>. Сейчас я могу представить себе, какое разочарование должен был испытать Андрей Петрович, когда отыскал статью японского ученого Ёсихико Футамуры, опубликовавшего эти соотношения еще в 1971 году. Примером профессиональной этики является тот факт, что Андрей Петрович назвал независимо открытые им соотношения проекциями Футамуры, несмотря на то, что последней проекции в той статье не было.

Уже в первых работах были предложены основные методы реализации смешанных вычислений:

1. *Частичные вычисления*, при которых все инструкции исходной программы, которые зависят только от доступных данных, выполнялись непосредственно, а остальные – редуцировались и помещались в остаточную программу.
2. *Генерирующее расширение* разбивает процесс построения остаточной программы на два этапа. На первом этапе все действия программы классифицируются на доступные и задержанные и порождается программа, которая получает на вход доступные данные исходной программы и генерирует остаточную. Важность статической классификации действий программы на доступные и задержанные будет осознана значительно позже и названа *анализом периода связывания* (binding time analysis, BTA).
3. *Трансформационный подход*, при котором остаточная программа получается из исходной путем применения последовательности так называемых редуцирующих преобразований. К ним относятся, например, протяжка констант, редукция выражений, редукции условного с константным условием, раскрутка цикла, удаление неиспользуемых вычислений и т.п.

Также достаточно быстро определились основные проблемы смешанных вычислений – проблема задержанного управления и связанная с ней проблема остановки смешанных вычислений. Андрей Петрович искусно замаскировал эти проблемы в своих первых работах по смешанным вычислениям и признавал это в ответ на мои конкретные вопросы. Потребовалось еще много лет исследований и экспериментов, чтобы от наброска идеи прийти к практической реализации, но в тот момент основная задача состояла в том, чтобы внедрить идею перспективности смешанных вычислений в сознание программистского сообщества, и с этой задачей Андрей Петрович справился просто блестяще. Десятки раз, на всевозможных конференциях и семинарах, в статьях и неформальных беседах, он рассказывал о проекциях Футамуры и методах смешанных вычислений, формируя новое направление в системном программировании. Наглядным примером может служить развернутая, с большим количеством цветных иллюстраций статья в популярном, но не специализированном на программировании журнале «В мире науки»<sup>2</sup> [Ерш 84].

<sup>1</sup> Говорят, что Андрей Петрович несколько часов рассказывал сыну Василию о своем открытии по междугороднему телефону. Аналогичную историю я слышал и про Валентина Федоровича Турчина, который также независимо открыл эти соотношения.

<sup>2</sup> Русская версия американского журнала Scientific American.

Именно за работы в области смешанных вычислений Андрей Петрович был удостоен престижной премии им. Крылова Академии наук СССР.

### **Владимир Иткин – алгебра смешанных вычислений**

Другие исходные позиции и иной подход к исследованиям смешанных вычислений был свойственен работам Владимира Эммануиловича Иткина. Он производил впечатление замкнутого, отрешенного от окружающего мира ученого, для которого строгость доказательства и теоретическая сила результатов значительно важнее их практической применимости – типичный пример «теоретика»<sup>1</sup>. Несколько первых его работ были совместными с А.П. Ершовым, но впоследствии их сотрудничество стало не таким тесным. Причиной этого стало высказывание Андрея Петровича о том, что в совместных работах ему принадлежит основная идея, а технические детали прорабатывает Иткин, и даже последовавшее уточнение – *лишь* основная идея и *все* технические детали, – по-видимому, не удовлетворило Владимира Эммануиловича. Несмотря на то что работал он изолированно, результаты этих работ давали прочную теоретическую и понятийную базу для других исследователей.

Отправной точкой в исследованиях В.Э. Иткина была теория схем программ, с ориентацией в первую очередь на императивные операторные программы над общей памятью. Наиболее активно использовалось понятие экспликатора – оператора, обеспечивающего требуемое состояние части памяти программы. В простейшем случае экспликатор выражается присваиванием переменным заданных значений.

Роль экспликаторов была двоякой: с одной стороны, будучи обычными операторами, они являлись частью программы, так что в любой момент можно было прекратить процесс смешанного вычисления и объявить текущую программу остаточной, а с другой – они рассматривались как текущие точки вычислений. В.Э. Иткин исследовал и обосновал различные схемы организации смешанных вычислений – сквозную, пунктирную, поливариантную и др.

Последние работы В.Э. Иткина носили философский характер: частичные вычисления интересовали его как фундаментальный процесс перехода от общего к частному.

Владимир Иткин трагически погиб в 1991 году – замерз зимой по дороге в церковь.

### **Борис Островский – специализация синтаксических анализаторов**

Для того чтобы сделать идею смешанных вычислений жизнеспособной, нужно было найти ей практическое применение. Именно такую тему Андрей Петрович предложил в качестве диссертационной своему аспиранту Борису Наумовичу Островскому. Задача состояла в автоматическом преобразовании универсального синтаксического анализатора для некоторого класса грамматик в анализатор, специализированный для конкретной грамматики [Остр87]. Очевидно, что специализация синтаксических анализаторов, хотя и имела самостоятельную ценность, была лишь первым шагом на пути к реализации проекции Фугамуры, поскольку универсальный синтаксический анализатор вполне можно рассматривать как интерпретатор, грамматику – как интерпретируемую программу, а входную строку – как ее данные. С этой точки зрения речь шла лишь о первой проекции и задача достижения самоприменимости смешанных вычислений не ставилась.

<sup>1</sup> Меня всегда поражала технология написания статей, которую использовал В.Э. Иткин, – своего рода Cut&Paste в бумажном варианте. Если ему требовалось вставить или переставить пару предложений в рукописном тексте, то он делал это не редакторскими пометками или текстом на обратной стороне страницы, а буквально вырезал и вклеивал кусок страницы с написанным текстом в нужное место. В результате получалась длинная бумажная лента, которая живописно занимала весь его кабинет. По завершению работы над рукописью оставалось только разрезать ее на части стандартного размера и отнести машинистке для перепечатки.

Смешанный вычислитель Островского был основан на трансформационном подходе. Смешанного вычислителя как такового не существовало, скорее речь шла о трансформационной машине со специально подобранным набором преобразований. Процесс применения преобразований был аналогичен процессу исполнения нормальных алгоритмов Маркова. Текст универсального анализатора размечался маркерами преобразований. Обычно в начале процесса имелся единственный маркер – начального преобразования, помещенный перед первой инструкцией программы. Далее недетерминированно повторялась итерация выбора маркера и применения преобразования к помеченной ею конструкции. Результатом преобразования являлась модификация программы и расстановка новых маркеров. Так повторялось до тех пор, пока оставался хотя бы один маркер.

Набор преобразований разрабатывался для каждого универсального синтаксического анализатора заново. Фактически получался отдельный смешанный вычислитель для каждого класса грамматик. Но даже такой подход технологически оправдан. Во-первых, переиспользуется набор базовых преобразований, что гарантирует корректность специализированного анализатора. Во-вторых, для данного класса грамматик требуется только один смешанный вычислитель, который, будучи один раз «отлажен», может применяться к большому количеству грамматик. Наконец, в-третьих, система содержала развитые средства преобразования грамматик, такие, например, как выделение регулярной части и средства пост-обработки, применяемые к остаточным программам.

Система смешанных вычислений, разработанная Б.Н. Островским, была в свое время одной из самых развитых. К сожалению, рутинная преподавательская работа в провинциальном вузе не позволила ему продолжить активную работу в этой области.

### **Летняя школа по смешанным вычислениям в Лиманчике**

Думаю, что всесоюзный уровень исследования в области смешанных вычислений получили после конференции по смешанным вычислениям, прошедшей на спортивной базе «Лиманчик» Ростовского государственного университета. Конференция прошла несомненно успешно – она дала возможность встретиться и завязать рабочие контакты всем активным исследователям в этой области. Видимо, не имеет смысла, да и возможности, обсуждать сейчас программу конференции – отмечу только два запомнившихся «рабочих» момента.

Виктор Николаевич Касьянов представил доклад, посвященный редуцирующим преобразованиям программ. Достаточно широкий набор этих трансформаций составлял существенную часть системы СОКРАТ, нацеленную на обработку Фортран-программ. Интересным аспектом системы являлось то, что преобразования опирались не только на собственно текст программы, но и на пользовательские аннотации – утверждения о программе, которые считались априорно верными и потенциально могли как уменьшить сложность необходимого анализа, так и увеличить применимость преобразований. В дискуссии по докладу слово взял Николай Николаевич Непейвода и в свойственной ему манере безапелляционно заявил, что не найдется ни одной программы, к которой описанные преобразования можно было бы применить. Позже выяснилось, что он хотел сказать, что ни один программист сознательно не напишет ветвление с тождественно истинным условием, но к автоматически порожденным программам применение подобных преобразований может быть весьма перспективно.

В дискуссии Святослав Сергеевич Лавров высказал сомнение в практической полезности проекций Футамуры для реализации промышленных трансляторов, поскольку они покрывают самую простую и хорошо проработанную синтаксически управляемую часть процесса. Наиболее интересные задачи трансляторной проблематики, такие как экономия регистров или исключение общих подвыражений, не имеют

истоков ни в смешанном вычислителе, ни в интерпретаторе и, значит, не могут появиться в автоматически получаемом трансляторе<sup>1</sup>.

### **Михаил Бульонков – поливариантные смешанные вычисления**

Идею того, как избежать заикливания при специализации интерпретатора, подсказала еще одна аспирантка Андрея Петровича – Татьяна Шапошникова. Она пришла в очную аспирантуру Вычислительного центра СО АН, будучи преподавателем программирования в НЭТИ (ныне НГТУ). Когда она в первый раз рассказала мне о своей идее, я был совершенно не готов ее воспринять, поскольку ее предложение приводило к использованию неструктурированных переходов goto. Опасность этого я видел в том, что использовавшиеся схемы смешанных вычислений существенно опирались на структурированность программ, например для «пронесения» доступной информации через задержанные условные операторы.

Мне удалось сформулировать достаточные условия остановки этого процесса: если, во-первых, анализ периода связывания классифицирует переменные и, следовательно, инструкции программы статически (т.е. так, что если переменная объявлена доступной, то она является таковой в любой момент специализации) и, во-вторых, множество значений, принимаемое доступными переменными, конечно, то любая ветвь поливариантной специализации неизбежно приведет к уже пройденному состоянию. В определенном смысле вычисления оказываются «запертыми» в матрице, строки которой соответствуют инструкциям исходной программы, а столбцы – состояниям доступной памяти. В каждой клетке этой матрицы находится соответствующая редуцированная инструкция.

Андрей Петрович с энтузиазмом воспринял идею этого метода<sup>2</sup> – она явилась ключевой в реализации самоприменимого смешанного вычислителя Mix, пригодного для выполнения всех трех проекций Футамуры. Для уменьшения «технических» проблем реализация была проведена для специально разработанного императивного языка программирования ЯР, среди базовых операций которого была, например, операция редукции выражения. Методологическому осмыслению полученных результатов посвящена наша совместная статья [БулЕрш86].

К сожалению, оказалось, что мы опять немного опоздали. Независимо, буквально на несколько месяцев раньше, успешная реализация всех проекций Футамуры для небольшого подмножества функционального языка LISP была выполнена Петером Сестофтом (Peter Sestoft) под руководством профессора Нила Джоунса (Neil Jones) из Института информатики университета Копенгагена (DIKU). В свою очередь, для них оказалось сюрпризом с запозданием обнаружить статью [Bul84], где описывалась техника, которую они независимо изобрели при реализации частичного вычислителя. Это стало началом длительного сотрудничества, взаимных визитов и здорового соперничества DIKU и новосибирской группы.

### **Гунтис Барздинь – частичные вычисления и фазы трансляции**

Гунтис Барздинь приехал в Новосибирск поступать в аспирантуру к Андрею Петровичу. Точнее, поступал он в аспирантуру ВЦ Латвийского университета, но хотел, чтобы Андрей Петрович был у него научным руководителем. Устроив свои аспирантские дела, Гунтис зашел ко мне и был весьма удивлен, узнав, что я лишь на несколько лет старше его. Еще до приезда в Новосибирск он читал мои статьи и представлял

<sup>1</sup> В более теоретической постановке близкую проблему позже сформулировал и исследовал Н. Джоунс [Jones88]: «Может ли специализация программ давать более, чем линейное ускорение?» Здесь предполагается, что измерение сложности учитывает только размер задержанной части данных, а размер доступных данных – константа.

<sup>2</sup> Именно Андрей Петрович перевел на английский язык (а честнее сказать – переписал на английском языке) мою статью [Bul84], которая потом многократно цитировалась.

меня солидным ученым-теоретиком, что на поверку не соответствовало действительности. В ходе наших обсуждений смешанных вычислений и проекций Футамуры Гунтис заметил, что в них используется только специализация программ и игнорируется возможность получения промежуточных данных. Было бы интересно посмотреть, как могли бы выглядеть проекции в более общей трактовке смешанных вычислений.

Пусть для простоты формального изложения смешанные вычисления представляются двумя процессорами: специализатора *spec*, порождающего остаточную программу  $p_x$ , и частичного вычислителя *peval*, формирующего промежуточные данные  $x_p$ :

$spec(p, x) = p_x$ ;  $peval(p, x) = x_p$  такие,

что  $p(x, y) = p_x(x_p, y)$  для любого  $y$ . С одной стороны, если *peval* – тривиальный процессор и  $x_p$  всегда пусто, то эти соотношения вырождаются в обычное уравнение смешанных вычислений. С другой стороны, если результат доступных вычислений находит свое выражение в промежуточных данных, то можно считать, что задача смешанных вычислений заключается в переводе данных  $x$  в некоторое, в каком-то смысле более эффективное, представление  $x_p$ .

Подставляя в указанные соотношения интерпретатор *int* вместо *p*, программу *p* вместо  $x$  и данные *d* программы *p* вместо  $y$ , получим

$spec(int, p) = int_p$ ;  $peval(int, p) = p_{int}$  такие,

что  $int(p, d) = int_p(p_{int}, d)$ .

Таким образом, кроме перевода программы в некоторое промежуточное представление смешанные вычисления порождают интерпретатор этого представления. В вырожденном случае порождение интерпретатора  $int_p$  вообще не зависит от программы *p*, а только от результатов анализа периода связывания. Такой интерпретатор отличается от исходного тем, что там, где исходный интерпретатор производил вычисления над исходными данными, остаточный интерпретатор извлекает результаты этих вычислений из структуры данных  $p_x$ . Естественно, что в остаточном интерпретаторе должны появиться вспомогательные переменные, которые служат для указания текущего элемента данных  $p_x$ .

Разработанный нами алгоритм частичного вычисления был зеркальным отражением поливариантной специализации: там, где при специализации порождался очередной фрагмент остаточной программы, частичный вычислитель порождал элемент промежуточных данных, помещая в него значения доступных выражений этого фрагмента.

В случае, когда доступные переменные могут быть разбиты на достаточно независимые части, появляется возможность смешанной стратегии, при которой одна часть доступных вычислений выражается в остаточной программе, а другая – в промежуточных данных. Получаемая в этом случае остаточная программа может быть при необходимости в свою очередь специализирована относительно промежуточных данных. В приложении к проблематике трансляции такая стратегия соответствует последовательным *фазам трансляции*.

Основное преимущество частичного вычисления заключается в том, что при том же самом объеме доступных вычислений, что и при специализации, результат получается более компактным, поскольку исчезает необходимость «декорировать» вычисленные значения фрагментами исходной программы. С другой стороны, платой за это стали расходы на остаточную интерпретацию [БарзБул88].

Мы не довели идею поливариантного частичного вычисления даже до экспериментальной реализации. Позже ее выполнила Каролина Малкмьяр из Копенгагенского университета.

Диссертацию Гунтис Барздинь писал по тематике индуктивного синтеза программ, которую успешно защитил уже после кончины Андрея Петровича.

### Дмитрий Кочетов – M2Mix

Научным руководителем Дмитрия Кочетова я стал по рекомендации Игоря Васильевича Поттосина. Как-то сразу определилась тема диссертации – смешанный вычислитель для реального языка программирования. Весьма поверхностные в тот момент знания Дмитрия о смешанных вычислениях не позволяли ему даже представить, за сколь сложную задачу он берется. В качестве входного языка была выбрана Модула-2, которая в то время была не просто весьма популярным в ИСИ СО РАН языком, но и являлась языком системы программирования, разрабатываемой под руководством И.В. Поттосина в рамках договора с большой промышленной организацией. Появлялась реальная перспектива действительно практического использования смешанного вычислителя<sup>1</sup>.

Проект M2Mix потребовал разработки нетривиальных теоретических и реализационных решений. В частности, наличие в языке указателей и массивов потребовало нетривиального анализа периода связывания и, как его составной части, *анализа синонимов*.

Во избежание избыточного дублирования кода был разработан оригинальный анализ конфигураций, цель которого состояла в определении для каждой точки программы той части переменных, которые существенны для специализации. Несмотря на то что деление переменных на доступные и задержанные не меняется во время специализации, тривиальное решение, состоящее в хранении и сравнении активных и достигнутых состояний памяти целиком, делает смешанный вычислитель практически неработоспособным как по времени, так и по памяти. Оптимизировать процесс работы смешанного вычислителя можно, удалив из рассмотрения

- переменные, которые не изменяются в рассматриваемом фрагменте программы (такой переменной является, например, внутреннее представление программы при специализации интерпретатора);
- переменные, которые можно вычислить на основе значений существенных переменных;
- переменные, которые не используются в рассматриваемом фрагменте, например «мертвые» переменные<sup>2</sup>.

Аналогично бесперспективной будет попытка сравнивать состояния в каждой точке исходной программы. Для того чтобы избежать закливания процесса специализации, достаточно рассматривать, например, только множество точек, «разрезающее» все циклы программы, в которых изменяется доступная память. Это решение, очевидно, не оптимально: проблема заключается в том, что малое количество контрольных точек может привести к дублированию кода, а большое – к увеличению накладных расходов.

В отличие от большинства предыдущих проектов, M2Mix был реализован как генерирующий расширитель. Такое решение было принято не в последнюю очередь из соображений эффективности. Но основная причина носила более фундаментальный характер: такая реализация позволяла гораздо проще обеспечить то, что выполнение операций программы в процессе специализации в точности совпадало с их нормальным выполнением. Поскольку у нас не было другого способа выполнить программу, кроме как путем

<sup>1</sup> Тут надо заметить, что смешанные вычисления применялись в промышленных проектах. В качестве примера можно привести работы Самохадина [Самоч82] из Ленинградского политехнического института по специализации низкоуровневых управляющих программ, работы Романовского [Ром95] из Института автоматики СО РАН по оптимизации программ машинной графики. Но в нашем случае цель была более амбициозная – сделать практически применимым универсальный смешанный вычислитель, который, подобно транслятору, «знает» не о конкретной области использования, а только о входном языке.

<sup>2</sup> Указанные условия неточны и служат лишь для демонстрации идеи. Например, переменная может не использоваться в данном фрагменте, но ее следует считать существенной, поскольку необходимо транзитивно передавать ее значение в другой фрагмент, где она используется.

трансляции ее имеющимся компилятором, то точно так же должны были выполняться как специализация, так и остаточная программа<sup>1</sup>.

Одной из специфических черт языка Модула-2 является то, что в нем нет неструктурированной передачи управления goto. С точки зрения формального определения методов анализа это обстоятельство было несомненным плюсом, но для генерации остаточной программы оно вызвало определенные затруднения. Был использован «стандартный» прием, моделирующий метки и переходы глобальным циклом с вложенным переключателем.

Неожиданно оказалось, что, несмотря на то что в остаточной программе остался минимум интерпретации, она работала в несколько раз медленнее, чем исходная. Проблема опять оказалась в том, что эффективность программы определяется не только количеством выполняемых операций языка высокого уровня, но и тем, насколько эффективно сумел реализовать эти операции компилятор. И в этом смысле компактная и ясная программа исходного интерпретатора оказывается значительно лучше оптимизируема, нежели остаточная программа с нелокальными передачами управления и запутанными зависимостями по данным. На решение этой проблемы нацелена *постоптимизация*, улучшающая программу как с помощью традиционных оптимизирующих преобразований, так и специальными трансформациями, которые учитывают то, что программа была порождена смешанным вычислителем, например: знание о природе переменной PC.

M2Mix был опробован на стандартном наборе примеров, таких как быстрое преобразование Фурье и универсальный лексический анализатор Lex.

По результатам этих работ Дмитрий Викторович Кочетов в 1995 году успешно защитил диссертацию «Эффективная специализация алголоподобных программ». В настоящее время он работает в компании Microsoft.

### Улучшение специализируемости

Таким образом, в определенном смысле сбылось предсказание Н.Н. Непейводи: у нас имелся один из самых мощных смешанных вычислителей, но не было программ, к которым их можно было бы успешно применить, поскольку, даже разрабатывая универсальную программу, программист не задумывается о том, насколько она пригодна для специализации.

Хорошим дополнением к дальнейшему усилению смешанного вычислителя могут стать средства, которые модифицируют исходную программу, улучшая ее специализируемость. Достаточно большой набор таких преобразований ввел Б. Н. Островский. В дипломной работе Юрия Александровича Баннова номенклатура таких преобразований была существенно расширена и, более того, были сформулированы условия целесообразности их применения. Как и Б.Н. Островский, Ю.А. Баннов занимался применением смешанных вычислений к синтаксическим анализаторам, но у последнего свобода маневра была ограничена фиксированным смешанным вычислителем M2Mix и фиксированным универсальным анализатором Yacc<sup>2</sup>.

Также на повышение специализируемости программ была нацелена дипломная работа Владимира Яковлевича Курляндчика «Поливариантный анализ периода связывания для функций высших порядков».

<sup>1</sup> Однажды мы долго пытались выяснить разницу в результатах некоторой вычислительной программы и ее специализированной версии. Оказалось, проблема заключалась в том, что опции компилятора, которые были установлены при трансляции генерирующего расширения, отличались от тех опций, с которыми транслировалась остаточная программа.

<sup>2</sup> В ходе выполнения работы был выполнен любопытный эксперимент, который еще раз наглядно убедил нас в работоспособности концепции. Надо было получить два специализированных анализатора: один получить автоматически из универсального, а другой – написать с нуля вручную. Не говоря уже о том, что в написанном вручную так и не удалось искоренить все ошибки, он был просто менее эффективен, чем полученный автоматически.

В отличие от традиционной для новосибирской школы ориентированности на императивные языки программирования, здесь предметом исследования были функциональные программы. Общая идея состояла в следующем: при обычном моновариантном анализе периода связывания некоторая переменная объявляется задержанной, если она задерживается хотя бы при одном вычислении. Разработанные методы позволяли преобразовать исходную программу так, чтобы сделать больше вычислений доступными, раскопировав при необходимости как код программы, так и данные.

Дипломные работы В.Я. Курляндчика и Ю.А. Баннова были признаны лучшими в соответствующие годы. В настоящее время они оба работают менеджерами в компаниях, занимающихся разработкой программного обеспечения.

### Постскриптум

В 1992 году в ИСИ СО РАН была образована научно-исследовательская группа смешанных вычислений, преобразованная в 1997 году в лабораторию смешанных вычислений. Непростые девяностые годы сильно повлияли на характер исследований, поскольку приоритет сместился к обеспечению финансирования. Экспериментальные работы, которые не могли быть непосредственно использованы в промышленных проектах, приходилось замораживать либо выполнять силами студентов. Это происходило на фоне стремительного развития вычислительных средств: менялись парадигмы и технологии программирования, в десятки и сотни раз повышалась производительность машин, совершенствовалась техника компиляции. Многие вещи, которые ранее казались характерными для смешанных вычислений, такие как поливариантность, раскрутка циклов, открытая подстановка процедур, сейчас являются общепринятыми в развитых компиляторах.

Однако нельзя сказать, что накопленный опыт в области специализации программ бесполезен. В качестве примера приведу совместный проект, который выполняется в настоящее время лабораторией смешанных вычислений, петербургской компанией ТЕРКОМ и американской компанией Relativity Technologies [Терком01]. Целью этого проекта является разработка средств для реинжиниринга и модернизации устаревшего программного обеспечения. Одной из ключевых задач в этом процессе является так называемое *извлечение бизнес-логики*. Среди разных подходов к решению этой задачи примечателен один, называемый *domain-based slicing*, который по сути состоит в специализации программы для конкретных значений переменных. Хотя этот метод наверняка непригоден для получения объектного кода из интерпретатора, он дает удовлетворительное решение для многих нетривиальных задач, возникающих из реального применения системы и которые вряд ли могли возникнуть в чисто «академических» постановках, например:

- допущение не одного, а множества или диапазона значений, которые могут поступать на вход программы;
- возможность негативной специализации – задании множества значений, которые не может принимать некоторая переменная;
- задание значений не на входе программы, а в некоторой точке, например там, где значения считываются из базы данных;
- специализация программ, состоящих из многих компонент и др.

### Библиографический список

[БулЕрш86] – Бульонков М.А., Ершов А.П. Как специальные конструкции трансляции могут порождаться универсальными процессами смешанных вычислений // Вычислительные системы. Новосибирск, 1986. Вып. 116: Прикладная логика. С. 47–66.

- [Bul84] – Bulyonkov M.A. Polyvariant mixed computation for analyzer programs // Acta Informatica. 1984. Vol. 21, Fasc. 5. P. 473–484.
- [Ерш84] – Ершов А.П. Смешанные вычисления // В мире науки. 1984. № 6. С. 28–42.
- [Itkin88] – Itkin V.E. An Algebra and Axiomatization System of Mixed Computation // Partial Evaluation and Mixed Computation. North Holland, 1988. P. 209–224.
- [Остр87] – Островский Б.Н. Управляемые смешанные вычисления и их применение к систематическому получению языково-ориентированных синтаксических анализаторов // Программирование. 1987. Т. 2. С. 56–67.
- [Самоч82] – Самочадин А.В. Оптимизатор структурированных микрокомпьютерных ассемблерных программ // Программирование микропроцессоров. Таллин, 1982. С. 89–99.
- [Jones88] – Jones N.D. Challenging Problems in Partial Evaluation and Mixed Computation // Partial Evaluation and Mixed Computation. North-Holland, 1988. P. 1–14.
- [БарзБул88] – Барздин Г.Я., Бульонков М.А. Смешанные вычисления как средство выделения фаз трансляции // Методы трансляции и конструирования программ. Новосибирск, 1988. С. 21–23.
- [Ром95] – Романовский А.В. Применение смешанных вычислений для решения задач трехмерной машинной графики: Автореф. дис. ... канд. техн. наук. Новосибирск, 1995.
- [Терком01] – Автоматизированный реинжиниринг программ / Под ред. А.Н. Терехова и А.А. Терехова. СПб.: СПбГУ, 2001.
- [Bul93] – Bulyonkov M.A. Polyvariant binding time analysis // Proc. of the AC Sympos. on partial evaluation and Semantics Based Program Manipulation. Copenhagen, 1993. P. 59–65.

## КОНЦЕПЦИЯ ПРЕДСТАВЛЕНИЯ АКАДЕМИЧЕСКОЙ ОРГАНИЗАЦИИ В СЕТИ ИНТЕРНЕТ

**П.В. Воронина**

Институт вычислительных технологий СО РАН  
Новосибирск, РФ  
*voronina@ict.nsc.ru*

Доклад представляет опыт разработки новой версии официального сайта ИВТ СО РАН. Особое внимание уделяется вопросам двуязычности сайта, его существованию как элемента распределенной системы научного сообщества СО РАН и актуализации содержащейся на сайте информации.

## ТРАНСПЬЮТЕРНЫЕ СИСТЕМЫ – НАЧАЛО СТАНОВЛЕНИЯ В РОССИИ ЭВМ С МАССОВЫМ ПАРАЛЛЕЛИЗМОМ

**А.И. Галушкин**

Научный центр нейрокомпьютеров НИИ автоматической аппаратуры  
Москва, РФ  
*neurocomputer@yandex.ru*

Информация о первых транспьютерах появилась в начале 80-х годов. Объективной причиной их появления явилась возможность на базе достигнутого уровня технологии микроэлектроники реализовать в од-

ном кристалле 16-разрядный (а несколько позже 32-разрядный) микропроцессор, память и 4 канальных адаптера. Работа с первыми зарубежными образцами транспьютерных систем в период с 1987 по 1990 год показала их эффективность и перспективность использования.

Для дальнейшего развертывания сферы применения транспьютерных систем начиная с 1990 года была поставлена работа по созданию совместно с Координационным центром по информатике и вычислительной технике Болгарской академии наук по созданию и поставкам российским предприятиям транспьютерных систем следующих типов:

1. Транспьютерный АРМ;
2. Транспьютерный АРМ средней производительности;
3. Транспьютерный АРМ высокой производительности;
4. Транспьютерный АРМ сверхвысокой производительности;
5. Транспьютерная графическая станция;
6. Транспьютерная система обработки изображений;
7. Транспьютерный АРМ с дисковой подсистемой.

В целом в период с 1991 по 1992 год было поставлено более 150 транспьютерных систем.

В докладе представлена краткая характеристика данных систем.

### **О транспьютерных системах**

Под ЭВМ с массовым параллелизмом в отличие от классических многопроцессорных ЭВМ с числом процессоров 2, 4, 8, 16 (как правило, не более) понимают ЭВМ с числом процессоров 4, 8, 10, 16, 32, 128, 256, 512, 1024 и более, в которых соблюдается принцип линейного (или почти линейного) роста производительности в зависимости от числа процессоров (физического объема или стоимости). Появление таких ЭВМ связано с разработкой в начале 80-х годов английской фирмой Inmos транспьютера T414. В отличие от обычных микропроцессоров в транспьютере на кристалле (рис. 1) были реализованы:

- 32-разрядный микропроцессор;
- внутрикристалльное ОЗУ объемом 2 Кбайт;
- четыре канальных адаптера.

Для разработки следующей версии транспьютера T800 (рис. 2) фирме Inmos понадобилось около пяти лет. В него были введены два блока:

- блок выполнения операций с плавающей запятой;
- внутрикристалльное ОЗУ объемом 4 Кбайт.

Преимуществами транспьютерной идеологии построения сверхвысокопроизводительных ЭВМ были также развитое программное обеспечение, система интерфейсных СБИС (канальные, графические, ввода изображений и др.), система проблемно-ориентированных СБИС (обработка сигналов, изображений, дисковые системы, реализация тригонометрических функций и т.п.). Интерфейсные и проблемно-ориентированные транспьютерные СБИС имели, как и транспьютеры, свойство каскадируемости с использованием стандартных каналов связи.

### **Транспьютерные системы общего назначения**

Наиболее распространенный вариант транспьютерной системы, позволявшей в те годы построить суперЭВМ из обычной ПЭВМ 286 представляет собой интерфейсную плату (фото 1) и материнскую плату (рис. 3 и фото 2) с возможностью установки на нее транспьютерных модулей размера 1 (фото 3) до 8 штук;

размера 2 до 4 штук и размера 4 до 2 штук (фото 4) в любой комбинации. Причем пиковая производительность максимальной конфигурации такой системы была равна 1,5 MFLOPS или 90 MIPS при достаточно малой производительности ПЭВМ. Уже в то время говорилось, что транспьютерные системы имеют универсальное применение для задач, допускающих распараллеливание алгоритмов. В составе транспьютерного рабочего места поставлялись системы программирования на следующих языках:

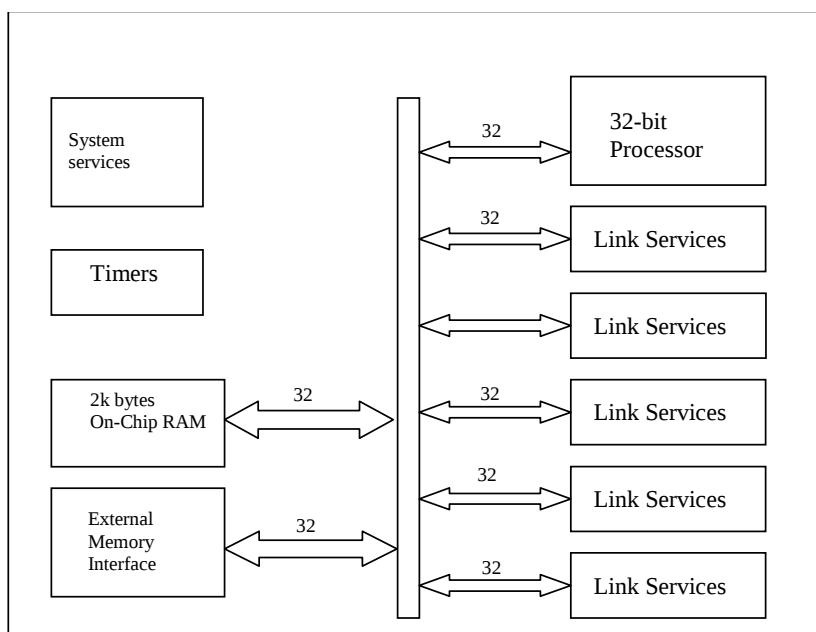
- Occam 2 (оригинальная среда TDS или среда MS-DOS);
- Parallel C;
- Parallel Fortran;
- Parallel Pascal;
- LISP, Prolog;
- C, Fortran, Pascal, Modula-2 в среде OC Helios (аналог Unix V);
- Ассемблер транспьютера.

} среда MS-DOS

На рис. 4 представлена схема транспьютерного рабочего места средней производительности, содержащего две платы – интерфейсную с одним транспьютером и базовую с четырьмя транспьютерами (фото 5). Пиковая производительность такой системы составляет 7,5 MFLOPS или 50 MIPS.

На рис. 5 представлена одна из наиболее популярных на период начала 90-х годов поставляемых конфигураций – транспьютерное рабочее место с интерфейсной транспьютерной платой и одной или несколькими платами, на каждой из которых размещалось 10 транспьютеров (фото 6) с локальной памятью 1 Мбайт. Разработчик мог иметь на столе рабочую станцию в несколько десятков транспьютеров. Для случая одной 10-транспьютерной платы производительность такого рабочего места достигала 16,5 MFLOPS или 110 MIPS. При этом необходимо отметить, что производительность наиболее массовых тогда ЭВМ серии ЕС достигала всего нескольких MFLOPS, а ЭВМ представляла собой сооружение из нескольких стоек, занимавших помещение значительной (несколько десятков квадратных метров) площади.

Отметим, что состав программного обеспечения, указанный выше для простейшей транспьютерной системы, сохранялся и для других перечисленных конфигураций.



**Рис. 1. Архитектура транспьютера T414**

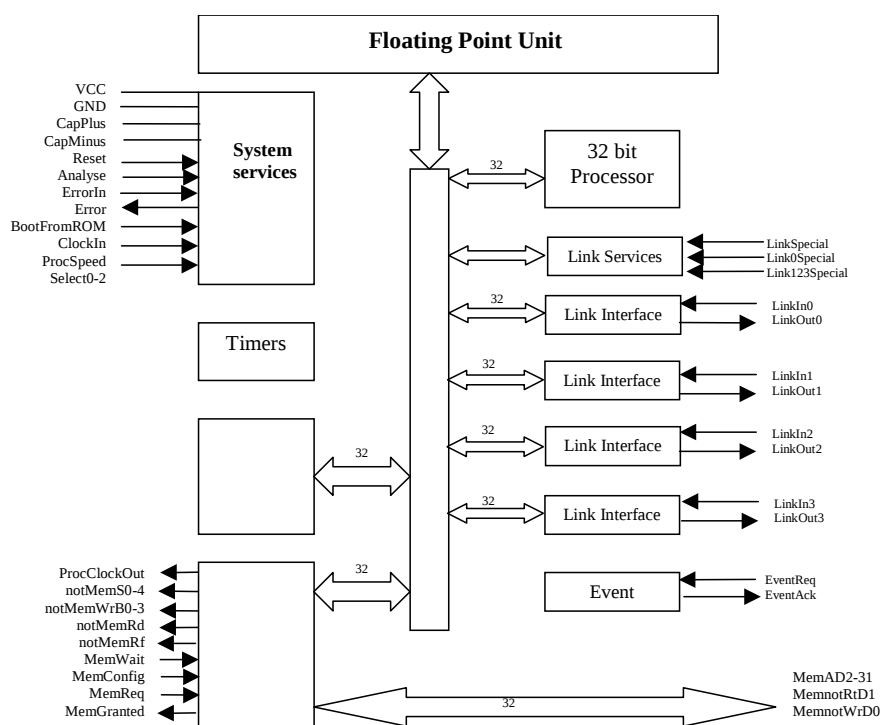


Рис. 2. Архитектура транспьютера T800

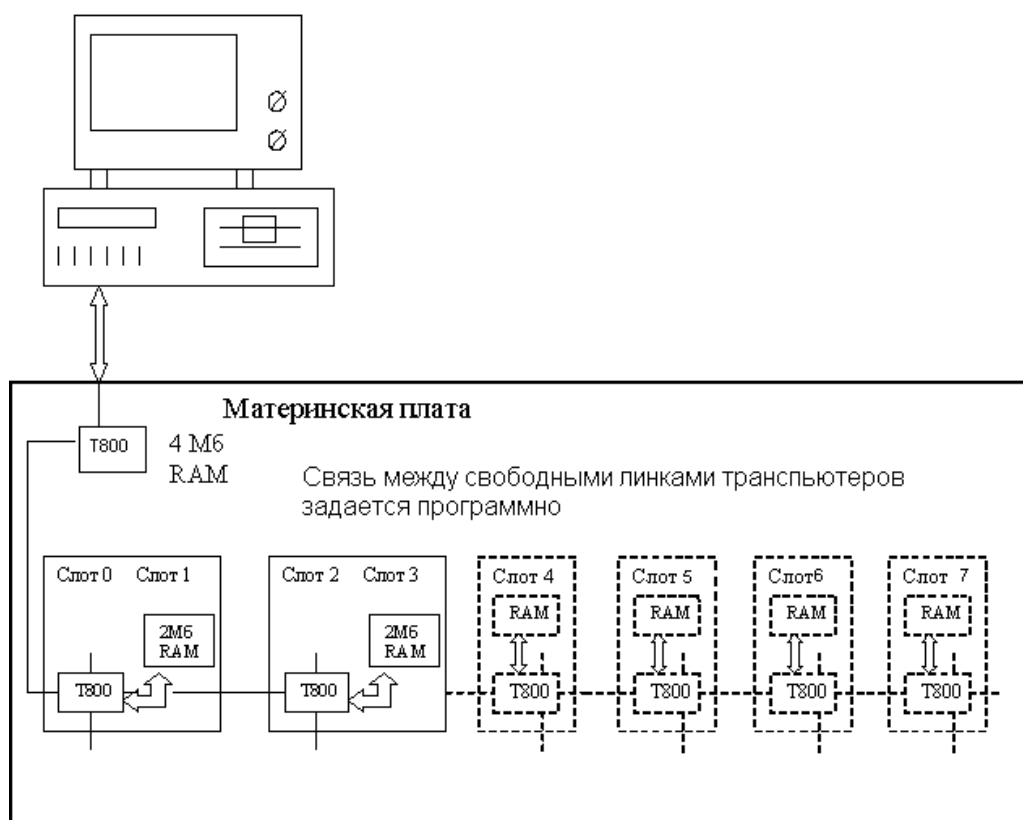
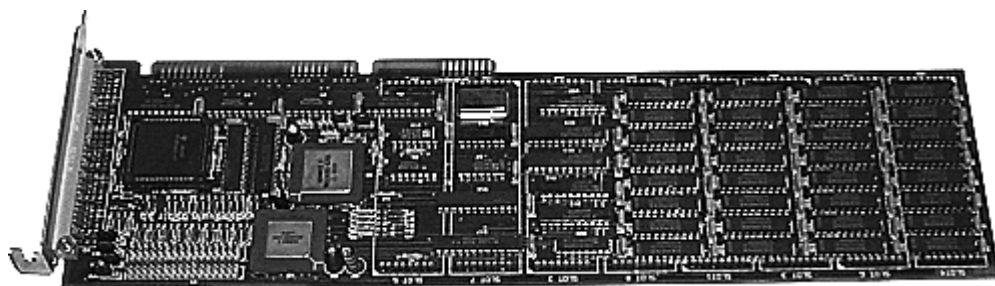
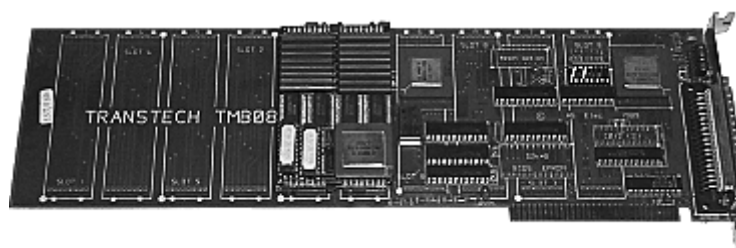


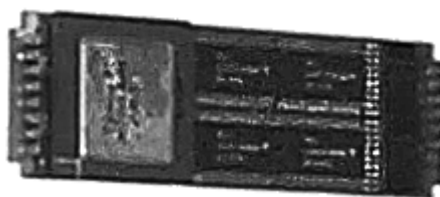
Рис. 3. Архитектура транспьютерного рабочего места с материнской транспьютерной платой и возможностью установки дочерних транспьютерных модулей



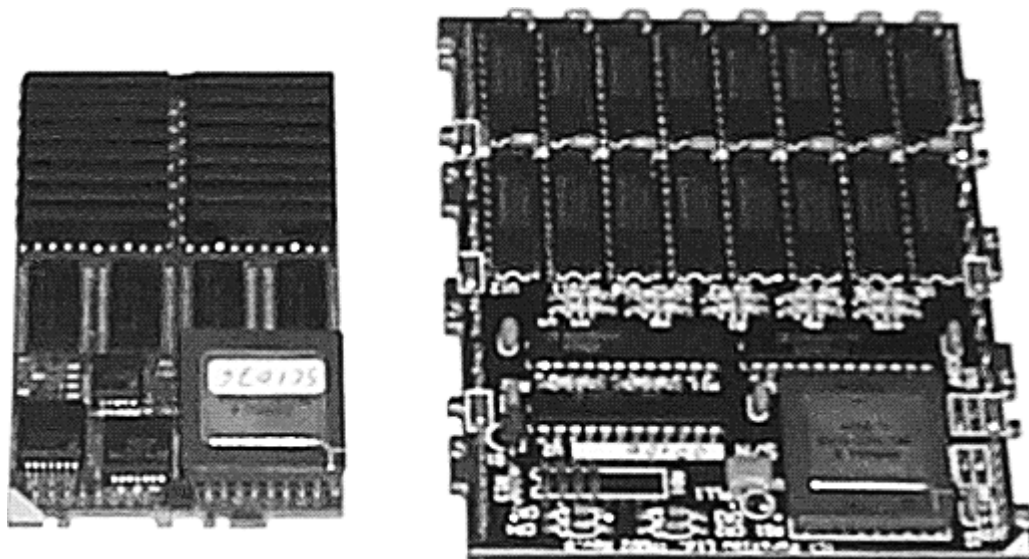
*Фото 1. Интерфейсная транспьютерная плата*



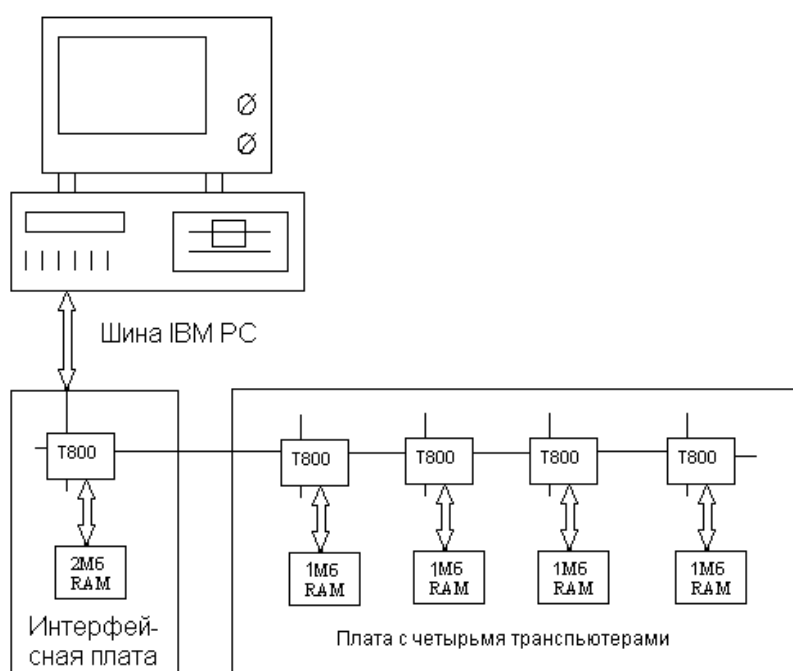
*Фото 2. Материнская транспьютерная плата*



*Фото 3. Транспьютерный модуль размером 1*



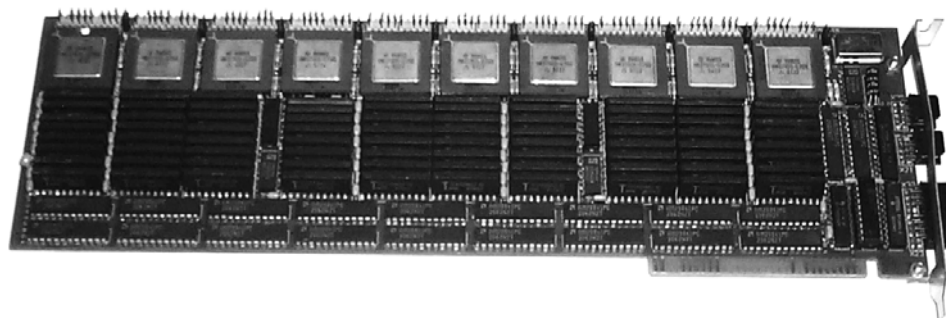
*Фото 4. Транспьютерные модули размером 2 и 4*



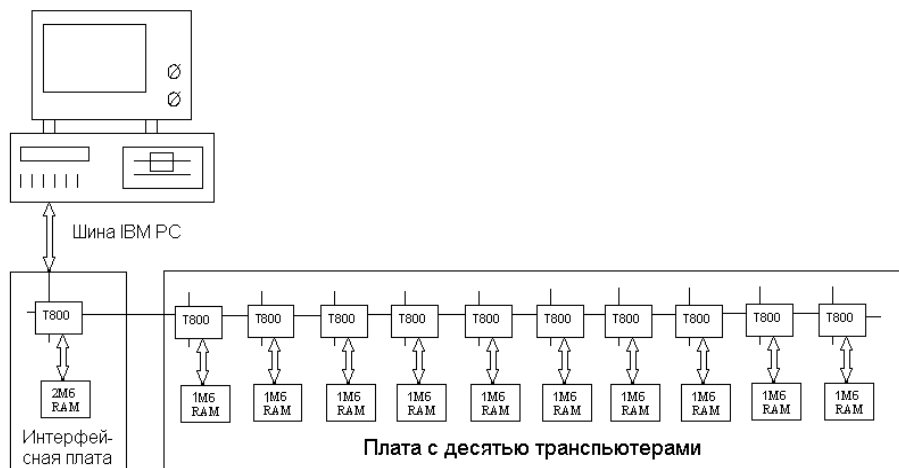
*Рис. 4. Архитектура транспьютерного рабочего места средней производительности*



*Фото 5. Транспьютерная плата с четырьмя транспьютерами*



*Фото 6. Транспьютерная плата с десятью транспьютерами*



**Рис 5. Архитектура транспьютерного рабочего места с одной или несколькими 10-транспьютерными платами**

### Специализированные транспьютерные системы

Характерной положительной особенностью транспьютерных систем, в настоящее время слабо реализуемой в современных кластерных суперЭВМ, являлась возможность реализации специализированных подклассов задач архитектур. К ним относились:

- транспьютерные графические станции;
- транспьютерные системы обработки изображений;
- транспьютерные системы обработки сигналов;
- транспьютерные дисковые подсистемы.

*Графические станции.* На рис. 6 представлена архитектура транспьютерной графической станции, в которой реализовались:

- программируемая топология;
- масштабируемая производительность;
- разрешение 1280 x 1024 и выше;
- цветовая палитра до 16106 оттенков (8, 18, 24 бита на пиксель);
- 25–30 кадров в секунду.

В состав программируемого обеспечения транспьютерной графической станции входили:

- ГРАФИЧЕСКИЕ БИБЛИОТЕКИ И СИСТЕМЫ для языков Parallel C, Parallel Fortran, Parallel Pascal;
- Сервер для работы с MS Windows;
- Система X-Window (ОС HELIOS).

Основные области применения подобных станций:

- графические системы реального времени: тренажеры, томография, системы виртуальной действительности;
- системы моделирования объектов, сред, процессов;
- различные САПР;
- научная визуализация: молекулярное моделирование, вычислительная гидродинамика;
- синтез реалистичных изображений, реклама, компьютерное кино.

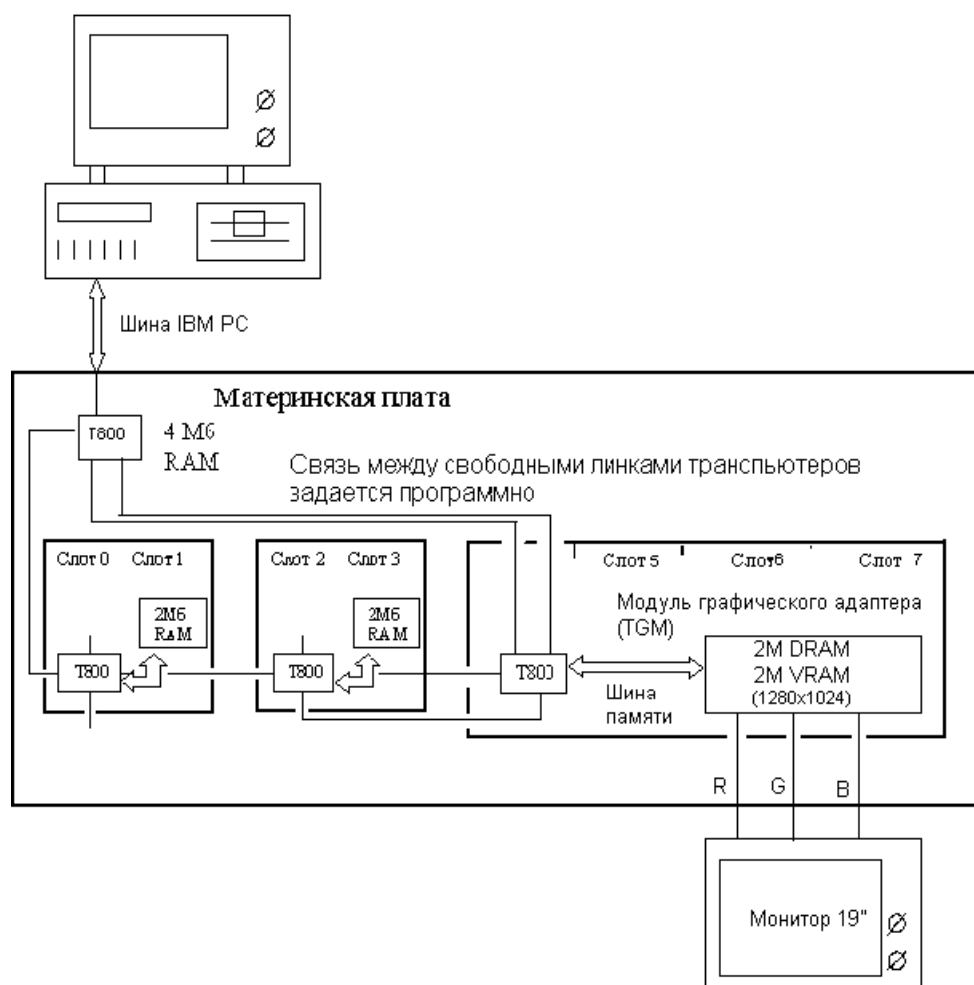


Рис. 6. Транспьютерная графическая станция

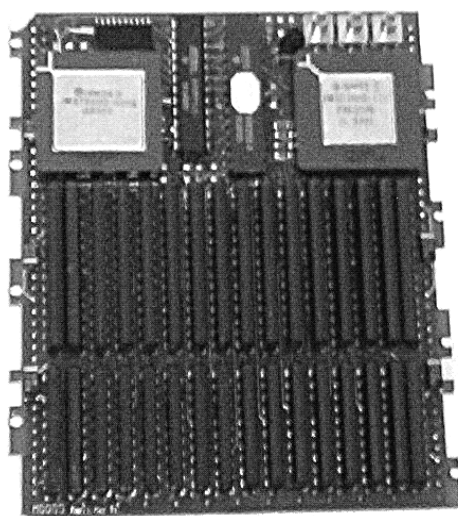


Фото 7. Транспьютерный модуль графического адаптера

*Транспьютерные системы обработки сигналов.* Принципиально важными и достаточно широко используемыми в начале 90-х годов прошлого столетия были специализированные транспьютерные рабочие станции обработки сигналов. Основу таких систем составлял специализированный транспьютер обработки сигналов IMS A100, схема которого представлена на рис. 7. Подобная аппаратная реализация каскадируемого Z-фильтра с перестраиваемой разрядностью весовых коэффициентов, с нашей точки зрения, представляет значительный интерес в настоящее время и в будущем с использованием текущей технологии микроэлектроники. На рис. 8, 9 представлены схема транспьютерного рабочего места обработки сигналов и схема платы APS-PC-DSP на базе сигнальных процессоров IMS A100 (фото 8). Отметим, что транспьютерное рабочее место обработки сигналов могло содержать несколько каскадируемых плат APS-PC-DSP. При наличии в рабочем месте только одной такой платы:

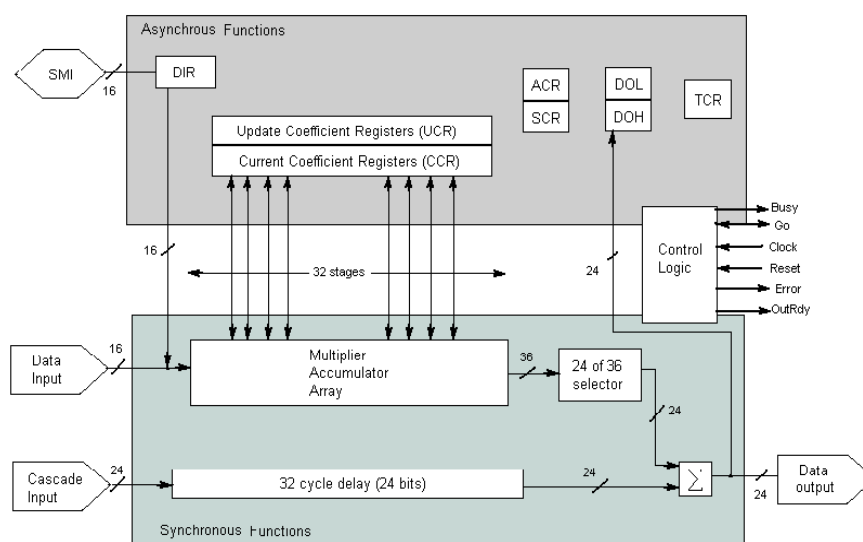
- производительность по обработке сигналов 1280 MOPS;
- максимальная скорость подачи входных сигналов до 10 Msamples/sec;
- производительность транспьютерной сети – 4,5 MFLOPS (30 MIPS),  
при закупке двух дополнительных TRAM – 7,5 MFLOPS (50 MIPS),  
при закупке 8 TRAM размера 1 – 13,5 MFLOPS (90 MIPS).

Программирование в рабочем месте осуществлялось на языке Оссам2 в среде TDS с помощью базовой системы DSP DS.

Были представлены пять примеров готовых программ по обработке сигналов.

Области применения:

- анализ сигнала, синтез формы сигнала;
- радарные, сонарные установки небольшой мощности;
- обработка, сжатие изображений, телевидение;
- обработка документов, сортировка писем, проверка багажа;
- распознавание рукописных и печатных символов;
- другие применения.



**Рис. 7. Структурная схема специализированного транспьютера обработки сигналов JMS A100**

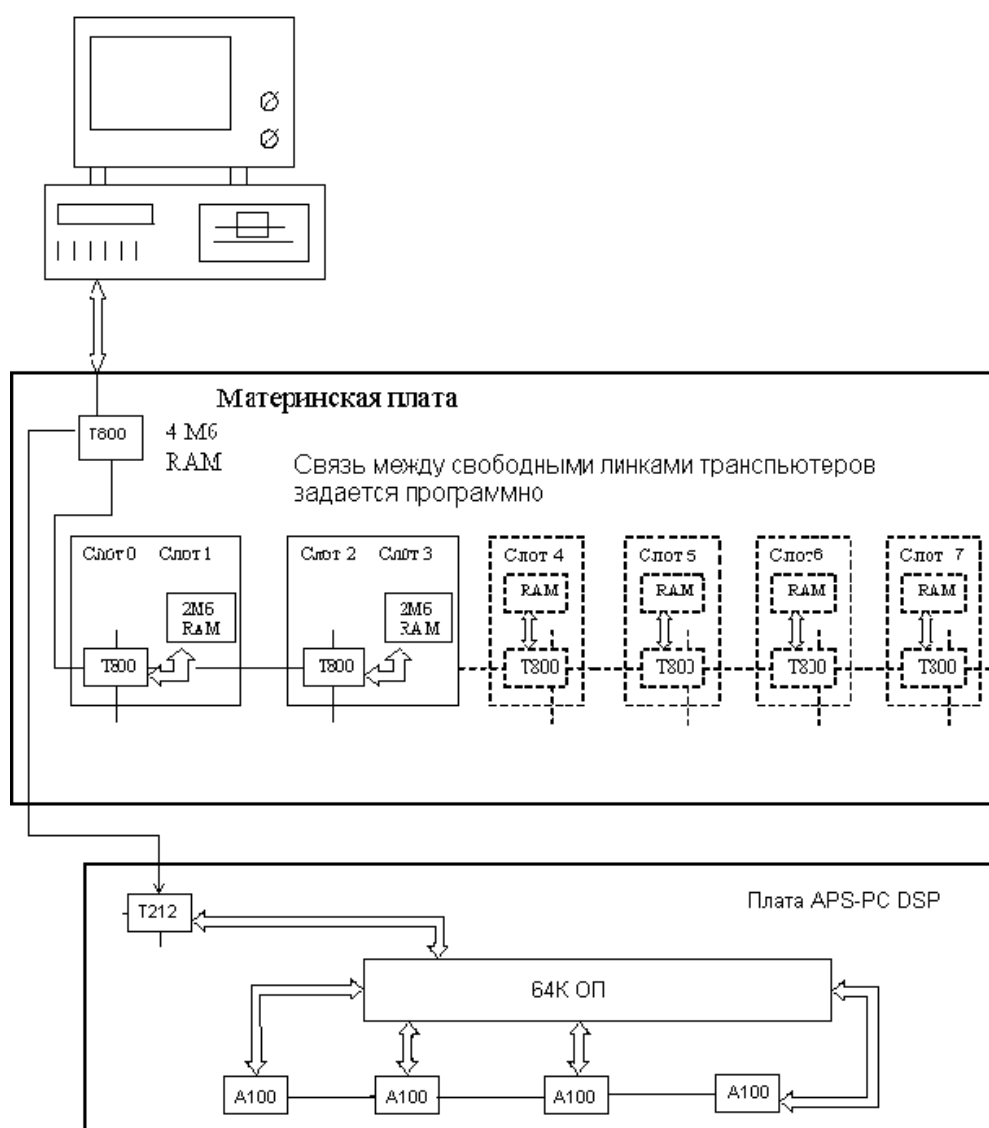


Рис. 8. Схема транспьютерного рабочего места обработки сигналов

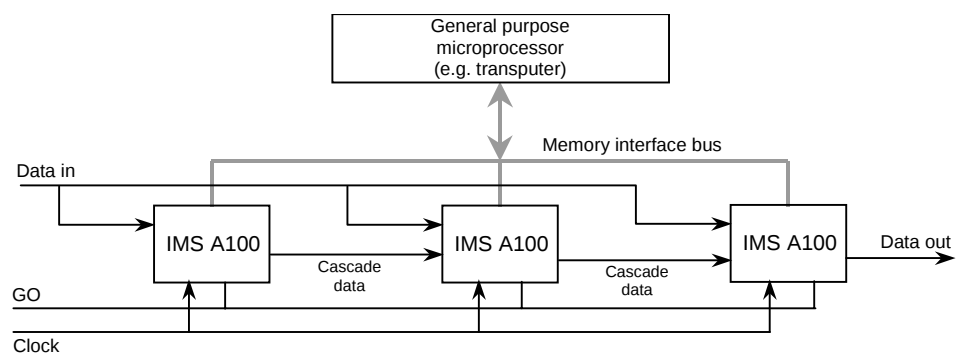


Рис. 9. Схема специализированной транспьютерной платы обработки сигналов

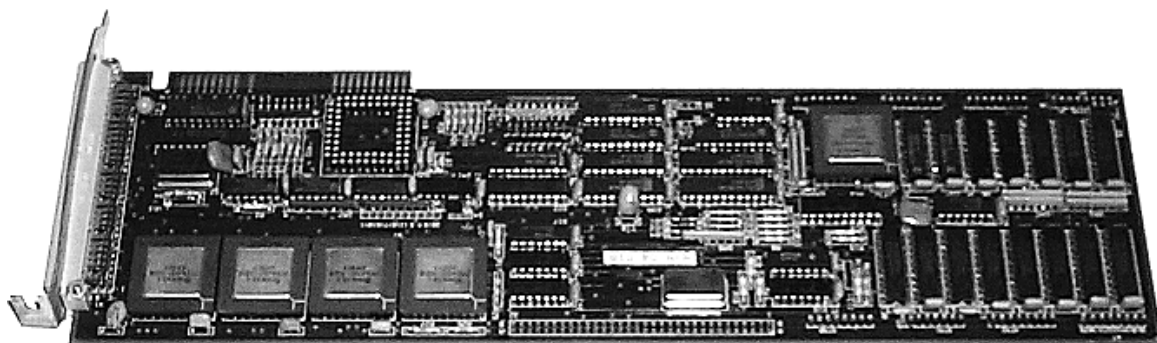


Фото 8. Специализированная транспьютерная плата обработки сигналов

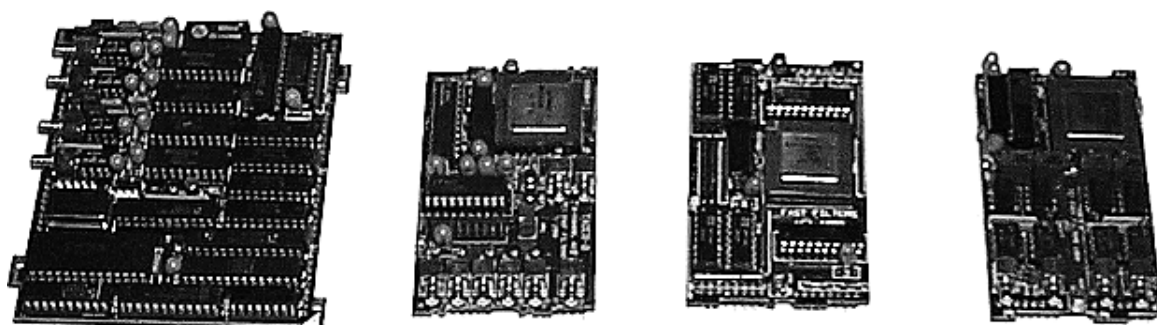


Фото 9. Транспьютерные модули АЦП и ЦАП сигналов

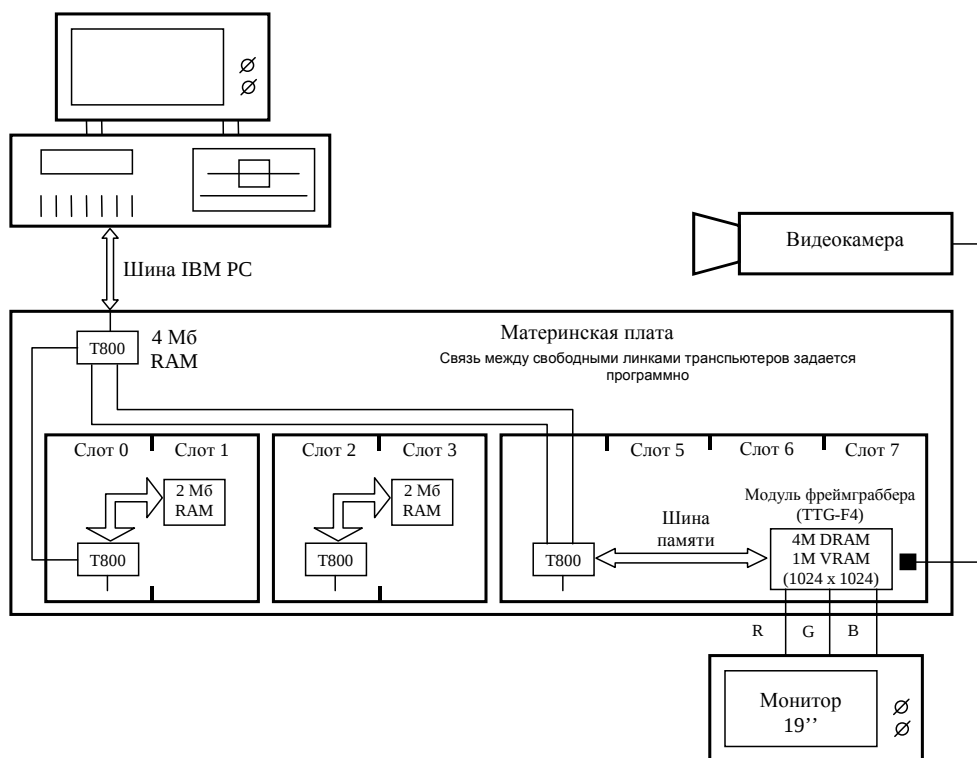


Рис. 10. Транспьютерная система обработки изображений

*Транспьютерные системы обработки изображений.* На рис. 10 представлена схема транспьютерной системы обработки изображений. На материнской плате размещается специальный транспьютерный модуль ввода изображений (фото 10).

Основными характеристиками транспьютерной системы обработки изображений являлись следующие:

- программируемая топология;
- шкалируемая производительность;
- разрешение 1024 x 1024;
- обработка изображений в реальном масштабе времени;
- «серые» и цветные изображения;
- высокие показатели «производительность/стоимость».

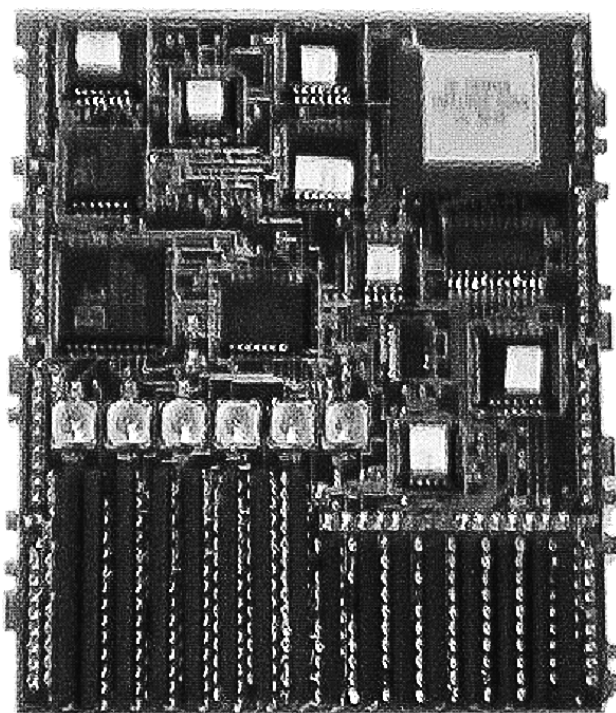
В составе системы поставлялись следующие программные средства:

- библиотеки функций обработки изображений IPLIB, IPAL для языков Parallel C, Occam 2, Parallel Fortran;
- управление конфигурацией TINY router, GENESIS program.

Данная система имела следующие области применения:

- системы технического зрения;
- промышленная автоматика;
- роботы;
- медицина, биология, металлургия и др.: анализ и обработка изображений;
- спутниковые системы наблюдения: экология, картографирование и др.

Более совершенная по сравнению с представленной на рис. 10 транспьютерная система обработки изображений имела в своем составе специализированные платы на базе СБИС IMS A110. На рис. 11, 12 представлены структурные схемы платы и СБИС IMS A110.



*Фото 10. Транспьютерный модуль ввода изображений*

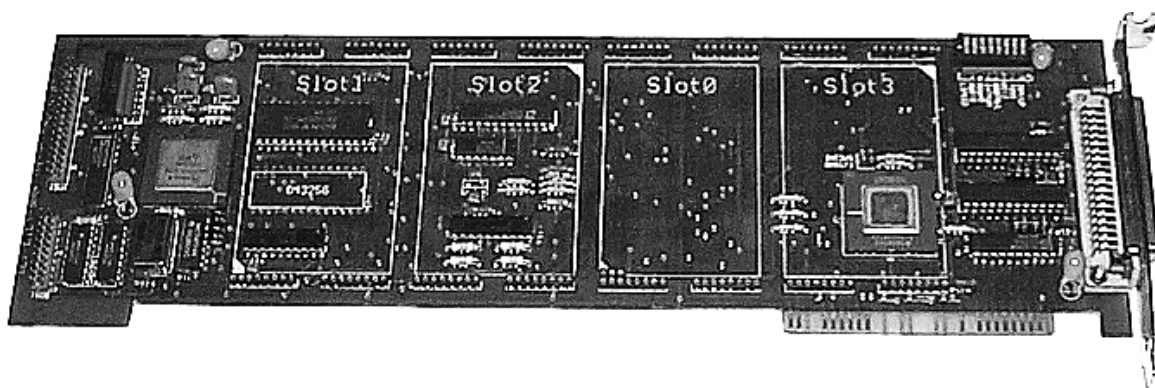


Фото 11. Транспьютерная плата, реализующая интерфейс дисковой системы

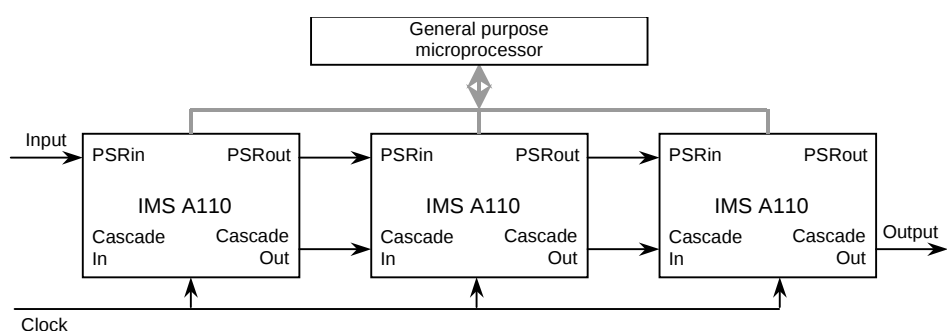


Рис. 11. Структурная схема транспьютерной платы обработки изображений на базе СБИС IMS A110

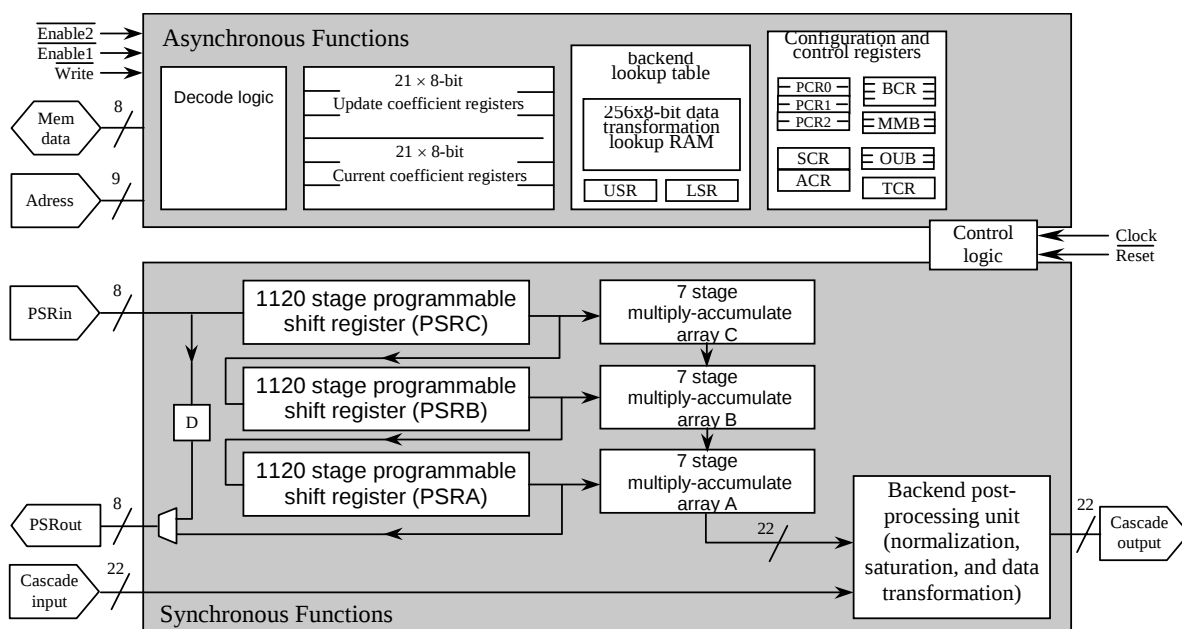
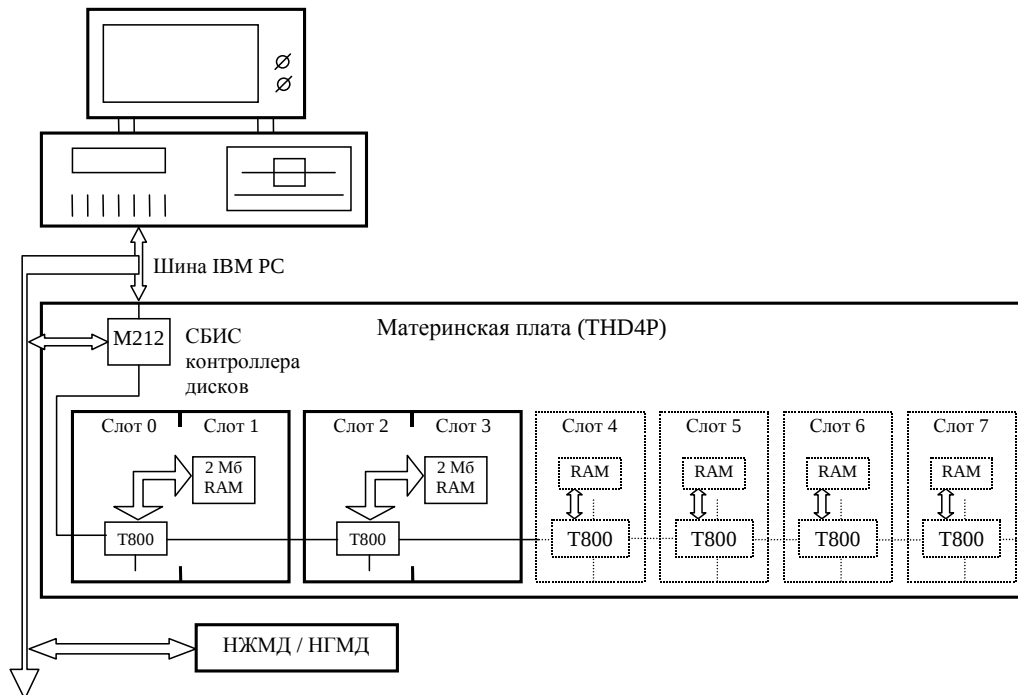


Рис. 12. Структурная схема СБИС IMS A110 – специализированного транспьютера обработки изображений



**Рис. 13. Структурная схема дисковой транспьютерной системы**

*Дисковые подсистемы.* Именно транспьютерные системы стали предвестником современных дисковых систем, реализующих RAID-массивы. Еще в начале 90-х годов прошлого столетия в России поставлялись дисковые транспьютерные системы, структура которых представлена на рис. 13. На фото 11 показана транспьютерная плата, реализующая интерфейс дисковой системы.

Основные характеристики подобной дисковой системы были в то время следующими:

- производительность транспьютерной сети при установке 8 ТРАМ с Т800 12 MFLOPS, 80 MIPS;
- интерфейс с шиной HOST-ПЭВМ (скорость обмена 200-300 Кбайт/сек); интерфейс с драйверами жестких и гибких магнитных дисков (до 4 дисков);
- емкость одного диска – 80 Мбайт.

Имелись в наличии следующие программные средства управления дисковыми операциями:

- TBIOS (Transputer BIOS) – библиотека процедур доступа к дисковым операциям M212 для языка программирования Оссам;
- TDOS – программная оболочка доступа к дисковым операциям M212 для системы программирования TDS;
- высокоуровневые файловые операции (типа open, read, write, close) для языков параллельного программирования (3L): C, Pascal, Fortran, обеспечивающие быстрый доступ к дисковым файлам через транспьютерные линки.

### **Транспьютерные рабочие станции сверхвысокой производительности**

Верхним уровнем коммерческих поставок транспьютерных систем в начале 90-х годов прошлого столетия были транспьютерные рабочие станции сверхвысокой производительности, схема которых представлена на рис. 14, а общий вид на фото 12.

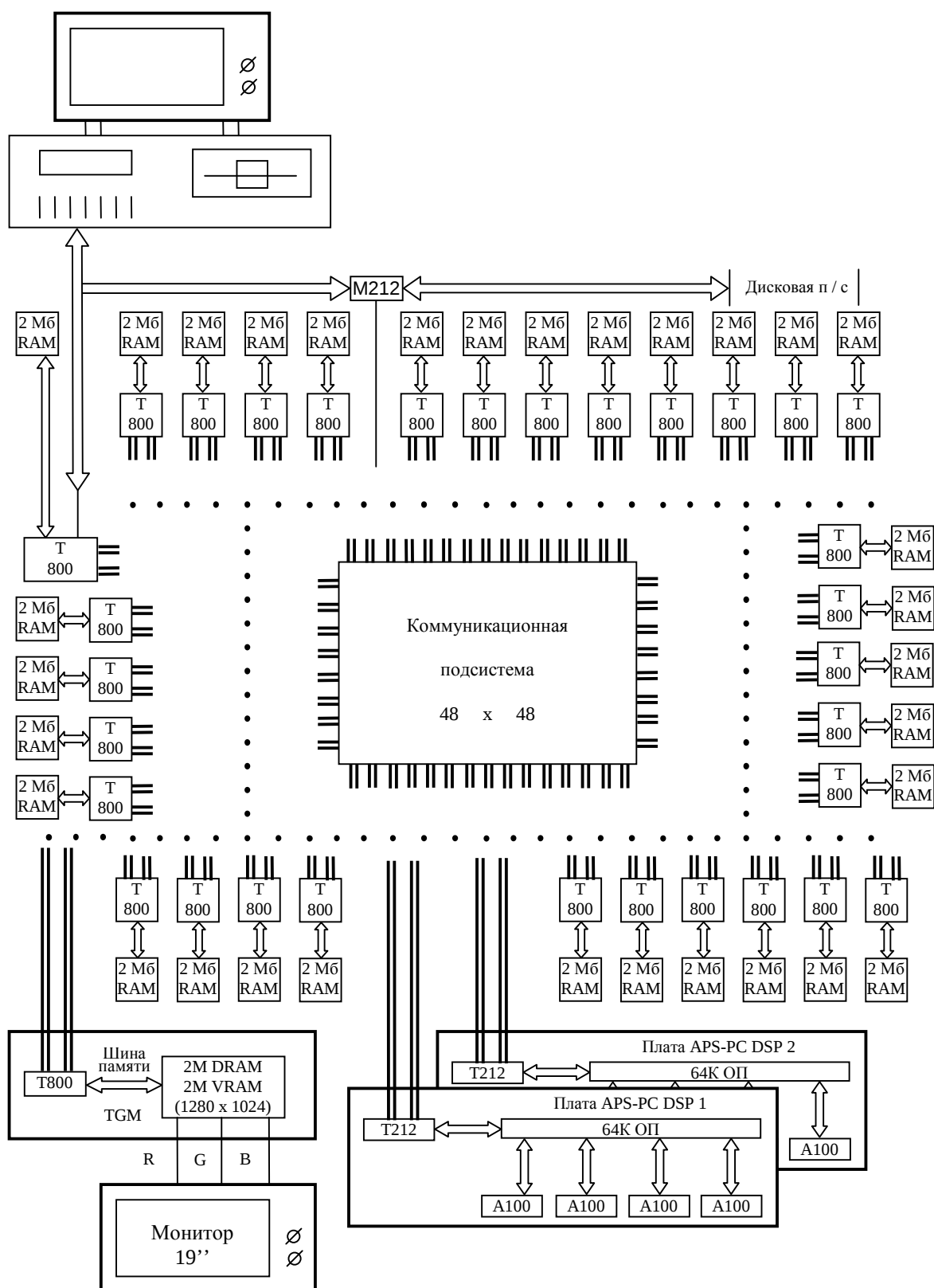


Рис. 14. Общая схема транспьютерной рабочей станции сверхвысокой производительности



**Фото 12. Общий вид транспьютерной рабочей станции  
сверхвысокой производительности**

Данный вариант рабочей станции состоял из пользовательской ПЭВМ, интерфейсных транспьютерных плат, массива транспьютеров с коммутационной системой, подсистемы обработки сигналов и дисковой подсистемы. Общая производительность рабочей станции определялась следующим образом:

|                                          |          |
|------------------------------------------|----------|
| Производительность транспьютерной сети   |          |
| 48 MFLOPS                                | 320 MIPS |
| Производительность по обработке сигналов |          |
| 2560 MFLOPS                              |          |
| Объем дисковой подсистемы – 320 Мб       |          |

Данные транспьютерные рабочие станции могли поставляться в любой комбинации специализированных транспьютерных подсистем.

Наряду с рассмотренной выше сверхвысокопроизводительной станцией на базе транспьютеров T800 (T805) предметом разработки и поставки в начале 90-х годов была сверхвысокопроизводительная рабочая станция на базе транспьютеров и процессоров i860. На рис. 15 представлена общая схема такой рабочей станции, а на фото 13 общий вид базовой платы с транспьютером и векторным процессором i860, суммарная пиковая производительность такой станции составляла 320MFLOPS. В составе рабочей станции поставлялось следующее программное обеспечение:

- PCi860-AL – пакет интерфейса стандартного ДОС к системе TTM100 включает компилятор языка C для i860;
- PCi860-C – компилятор языка C для i860;
- PCi860-F – компилятор языка ANSI FORTRAN 77 для i860;
- PCi860-M – пакет для управления из ПК неоднородной сетью процессоров IMST805 и i860 включая Inmos ANSI с Toolset IMSD7214;
- PCi860-D – символический отладчик для i860 на уровне командной строки.

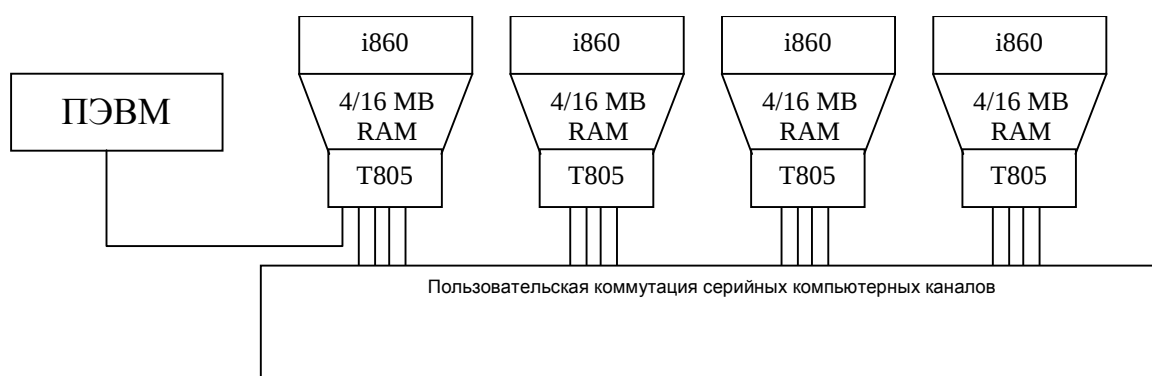
*Арифметические библиотеки для i860:*

i860lib-MA – математическая библиотека векторной обработки, включающая 260 функций;

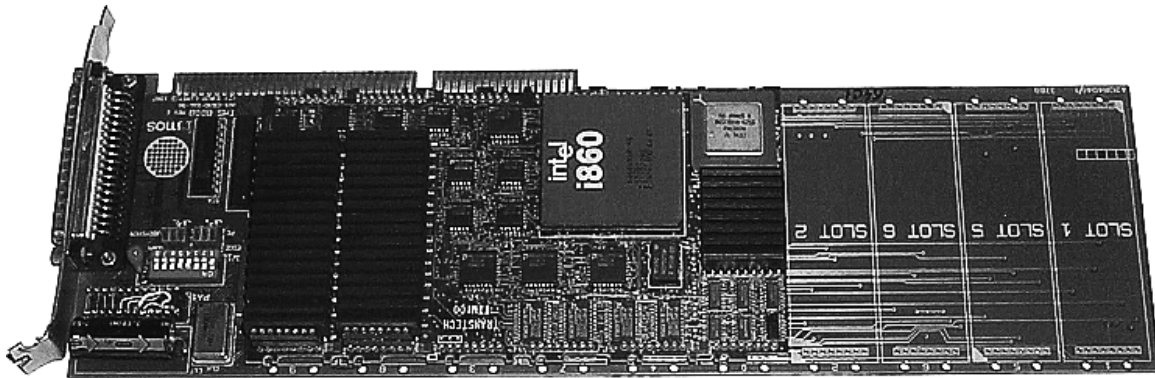
i860lib – IPLIB – библиотека функций обработки образов по стандарту NEL.

Необходимо отметить российскую разработку сверхвысокопроизводительной рабочей станции на базе транспьютерной платы в конструктиве «Эльбрус», содержащей 32 транспьютерных модуля с транспьютерами T800 (T805) и объемом памяти 1-4 Мбайта (фото 14).

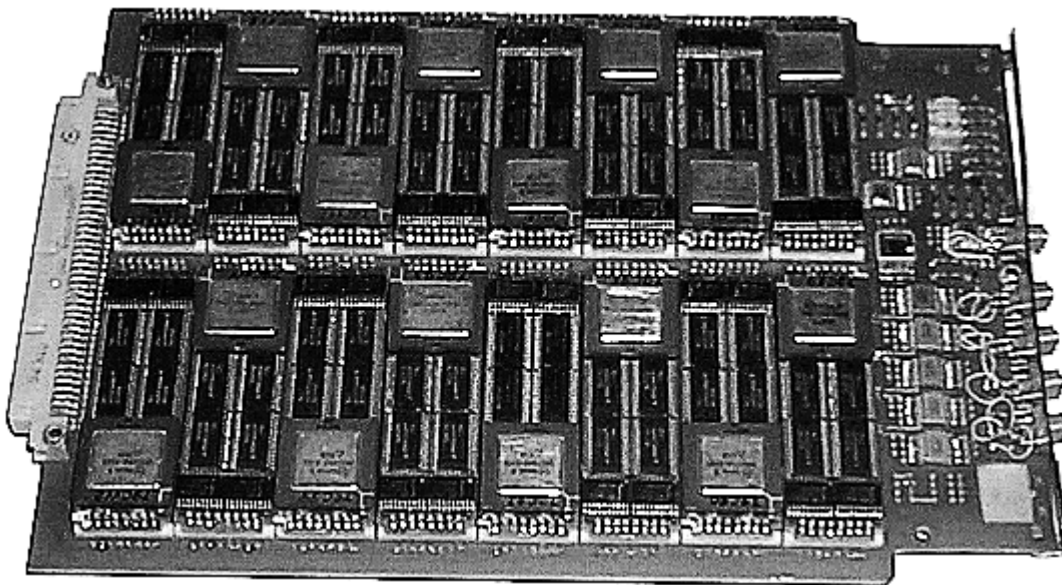
Транспьютерные модули на плате размещались с двух сторон платы по 16 модулей с каждой стороны. Пиковая производительность одной платы достигала 320MIPS, а блок в составе 8 плат имел пиковую производительность 2560MIPS. В начале 90-х годов вычислительная система в несколько миллиардов операций в секунду, да еще в виде настольного блока была уникальной.



**Рис. 15. Общая структура сверхвысокопроизводительной рабочей станции на базе транспьютеров и векторных процессоров i860**



*Фото 13. Транспьютерная плата с векторным процессором i860*



*Фото 14. Общий вид отечественной транспьютерной платы  
(32 транспьютерных модуля)*

### **Транспьютерные системы на базе транспьютера T9000**

В середине 90-х годов фирма Inmos, испытывая материальные и технологические трудности в разработке следующего поколения транспьютеров, вошла в консорциум Томсон CSF (Франция) и стала изготавливать перспективные транспьютеры на заводе в Гренобле, где, по всей видимости, имелись большие технологические возможности по сравнению с прежним производством в г. Бристоль (Англия).

Очередное поколение транспьютеров представлял транспьютер T9000 (рис. 16), образцы которого с малой (25 МГц) тактовой частотой работы были изготовлены. На основе этих транспьютеров было создано некоторое количество высокопроизводительных систем, однако фирме Томсон CSF так и не удалось разработать образцы транспьютера T9000 с более высокой тактовой частотой и большой скоростью передачи информации по каналам связи между транспьютерами. Было изготовлено небольшое количество образцов систем на базе транспьютера T9000 (фото 15), которые так и не пошли в серийное производство.

Транспьютеры в течение многих лет являлись основой построения сверхвысокопроизводительных ЭВМ во многих странах мира: Англии, Германии, Франции, США, Японии, Китае, России, Болгарии и др.

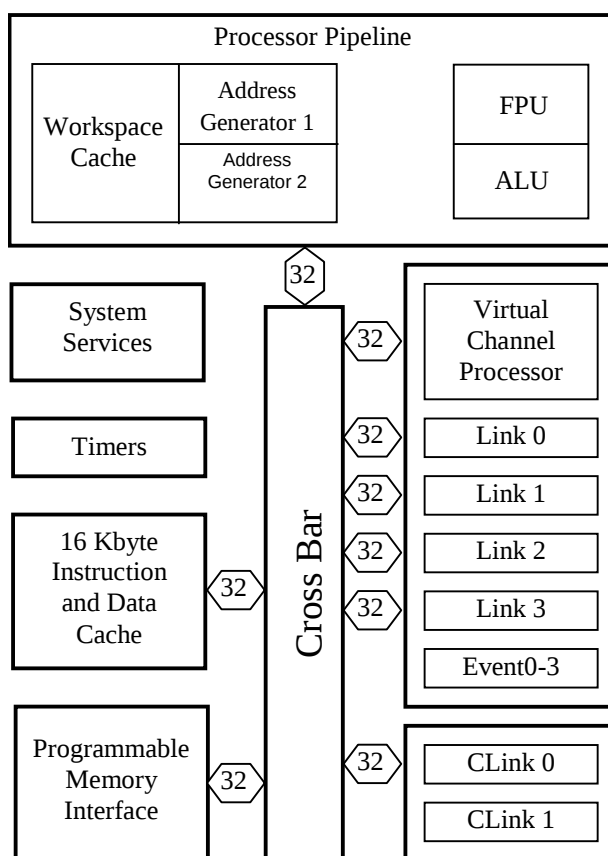


Рис. 16. Общая структура транспьютера T9000



Фото 15. Общий вид транспьютерной системы на базе транспьютера T900

## Заключение

Однако за последние годы структура рынка транспьютеров изменилась: вследствие конкурентной борьбы было целостное подавление европейской инициативы по ликвидации монополии США в области суперЭВМ. Для этого в конце 80-х годов США была развернута разработка по крайней мере трех эффективных транспьютероподобных элементов для ЭВМ с массовым параллелизмом: TMS32C40 (C60, C80), SHARK и «Neuron», по архитектуре близких к транспьютерам и выполняющих практически те же функции, но обладающих большими возможностями.

Необходимо отметить, что фактически транспьютеры явились во всем мире, в том числе и в России, началом становления ЭВМ с массовым параллелизмом. Современная кластерная суперЭВМ отличается от транспьютерных систем лишь типом процессоров, объемом памяти и некоторыми деталями архитектурных построений.

Еще в начале 90-х годов уникальные транспьютерные системы состояли из нескольких сотен транспьютерных узлов и по тем временам представляли собой самые мощные суперЭВМ по критерию отношения производительности к стоимости (или объему оборудования). Необходимо также отметить, что транспьютерные системы были в то время наиболее эффективными программно-аппаратными эмуляторами-ускорителями нейросетевых алгоритмов решения различных задач.

## Библиографический список

1. Галушкин А.И., Грачев Л.С., Точенов В.А. Алгоритмы реконфигурации в многомикроспроцессорных системах // Кибернетика. 1990. № 2.
2. Транспьютерные ЭВМ. М.: Изд-во НИИ «Квант», 1990.
3. Справочник по транспьютерным системам / А.И. Галушкин, Г.С. Елизаров, В.А. Точенов, К.К. Хачатурян. М.: Изд-во МАИ, 1990.
4. Galushkin A.I. The Architectures of Neural Computers // Joint British-Soviet Workshop on transputer systems, Moscow, 26–29 June, 1990.
5. Galushkin A.I. The Neural Computers on the Base of transputers and Signal processing. Moscow, 26–29 June, 1990.
6. Galushkin A.I., Tochenov V.A. and all. The pattern recognition systems with variable structure on the base of personal computer and transputer system // Joint British-Soviet Workshop on transputer systems, Moscow, 26–29 June, 1990.
7. Galushkin A.I., Kuzmin L.A. and all. The organization studying and improvement of professional skill of transputer system // Joint British-Soviet Workshop on transputer systems, Moscow, 26–29 June, 1990.
8. Алеутдинов М.А., Галушкин А.И., Медовый В.С. Система цитологических изображений на базе транспьютеров // Вторая Всероссийская конференция «Транспьютерные системы и их применение», 1992.
9. Galushkin A.I., Grachev L.V., Kravchenko V.V. A transputer network-based general purpose neurocomputer // Join Conference on Neural Networks, Nagosa, Japan 25 Oct., 1993.
10. Кузьмин Л.А. Курс лекций по языку ОККАМ-2. М., 1994.
11. Галушкин А.И. Нейрокомпьютеры. М., 2000.
12. Галушкин А.И. Некоторые исторические аспекты развития элементной базы вычислительных систем с массовым параллелизмом (80–90-е годы) // Нейрокомпьютер. 2000. № 1.

## ИНДЕКС ФАМИЛИЙ АВТОРОВ СТАТЕЙ

### А–Г

|                          |     |
|--------------------------|-----|
| Ahremchik O.L. ....      | 112 |
| Bojanov K.L. ....        | 152 |
| Bogoyavlenskiy Y.A. .... | 45  |
| Fet Ya.I. ....           | 99  |

### А–Б

|                         |              |
|-------------------------|--------------|
| Агамирзян И.Р. ....     | 112          |
| Александриди Т.М. ....  | 10, 15       |
| Алексеева З.Д. ....     | 30           |
| Бадрутдинова М.Ш. ....  | 115          |
| Белов С.Д. ....         | 120          |
| Берс А.А. ....          | 120, 128     |
| Богданова И.Ф. ....     | 135          |
| Богданова Н.Ф. ....     | 134          |
| Богоявленский Ю.А. .... | 33, 136, 145 |
| Бородин А.А. ....       | 145          |
| Боюн В.П. ....          | 152          |
| Брусенцов Н.П. ....     | 45, 153      |
| Бульонков М.А. ....     | 159          |

### В–И

|                       |     |
|-----------------------|-----|
| Владимирова Ю.В. .... | 153 |
| Воронина П.В. ....    | 168 |
| Галущкин А.И. ....    | 168 |
| Ефимов Г.Б. ....      | 51  |
| Захаров В.Н. ....     | 52  |
| Зуева Е.Ю. ....       | 51  |
| Иткин В.Э. ....       | 161 |

### К

|                        |     |
|------------------------|-----|
| Карпилович Ю.В. ....   | 72  |
| Климанов В.П. ....     | 53  |
| Корзун Д.Ж. ....       | 136 |
| Королев Л.Н. ....      | 61  |
| Криворученко В.С. .... | 62  |
| Крышень М.А. ....      | 136 |

|                   |     |
|-------------------|-----|
| Кулаков К.А. .... | 136 |
|-------------------|-----|

### Л

|                    |    |
|--------------------|----|
| Лавренюк Ю.А. .... | 15 |
| Левин И.И. ....    | 63 |
| Лукин Н.А. ....    | 63 |

### М

|                         |    |
|-------------------------|----|
| Малашевич Б.М. ....     | 64 |
| Малиновский Б.Н. ....   | 72 |
| Малиновский Л.Б. ....   | 67 |
| Митропольский Ю.И. .... | 67 |

### Н–П

|                          |     |
|--------------------------|-----|
| Никольская Ю.Н. ....     | 71  |
| Никульцев В.С. ....      | 120 |
| Пржиялковский В. В. .... | 72  |

### Р

|                     |                |
|---------------------|----------------|
| Рабинович З.Л. .... | 72             |
| Рогачев Ю.В. ....   | 10, 15, 21, 82 |
| Рузанова Н.С. ....  | 96             |
| Рябов Г.Г. ....     | 97             |

### С

|                        |    |
|------------------------|----|
| Семенов О.И. ....      | 97 |
| Смирнов Г.Д. ....      | 72 |
| Смолевичская М.Э. .... | 98 |
| Сутягин М.В. ....      | 53 |

### Ф–Щ

|                            |     |
|----------------------------|-----|
| Федотов А.М. ....          | 120 |
| Хетагуров Я.А. ....        | 30  |
| Хосе, Рамиль Альварес .... | 45  |
| Чубаров Л.Б. ....          | 120 |
| Шириков В.П. ....          | 105 |
| Щенков И.Б. ....           | 51  |

## ИНДЕКС НАИМЕНОВАНИЙ ОРГАНИЗАЦИЙ

|                                                                                                                        |               |                                                                                        |                  |
|------------------------------------------------------------------------------------------------------------------------|---------------|----------------------------------------------------------------------------------------|------------------|
| Institute of Computing Mathematics and Mathematical Geophysics of Siberian Branch of Russian Academy of Sciences ..... | 99            | МАДИ (ГТУ).....                                                                        | 10, 15           |
| Institute of Parallel Processing of Information BAN.....                                                               | 152           | Майкрософт Россия и СНГ .....                                                          | 112              |
| Petrozavodsk State University .....                                                                                    | 45            | МГТУ «Станкин» .....                                                                   | 53               |
| Tver State Technical University .....                                                                                  | 112           | МИФИ.....                                                                              | 30               |
| Институт вычислительных технологий СО РАН.....                                                                         | 168           | Московский государственный университет .....                                           | 45, 61, 153      |
| Институт кибернетики                                                                                                   |               | Музей истории КПО ВС .....                                                             | 115              |
| им. В.М. Глушкова НАН Украины .....                                                                                    | 67, 72, 152   | Научный центр нейрокомпьютеров НИИ автоматической аппаратуры .....                     | 168              |
| Институт машиноведения УрО РАН .....                                                                                   | 63            | НИИ вычислительных комплексов                                                          |                  |
| Институт подготовки научных кадров НАН Беларуси.....                                                                   | 134, 135      | им. М.А. Карцева .....                                                                 | 21, 82           |
| Институт прикладной математики                                                                                         |               | НИИ многопроцессорных вычислительных систем им. академика А.В. Каляева .....           | 63               |
| им. М.В. Келдыша РАН .....                                                                                             | 51            | ОАО «Ангстрем».....                                                                    | 64               |
| Институт проблем информатики РАН.....                                                                                  | 52, 71        | ОАО «НИЦЭВТ».....                                                                      | 72               |
| Институт систем информатики                                                                                            |               | Объединенный институт ядерных исследований....                                         | 105              |
| им. А.П. Ершова СО РАН.....                                                                                            | 120, 128, 159 | ОИПИ НАН Беларуси.....                                                                 | 97               |
| Институт системного анализа РАН .....                                                                                  | 67            | Петрозаводский государственный университет .....                                       | 33, 96, 136, 145 |
| Институт точной механики и вычислительной техники РАН .....                                                            | 97            | Политехнический музей .....                                                            | 98               |
| Институт ядерной физики им. Будкера СО РАН .....                                                                       | 120           | ФГУП «Центральный аэрогидродинамический институт им. профессора Н.Е. Жуковского» ..... | 62               |
| Концерн «Моринформсистема – Агат» .....                                                                                | 30            |                                                                                        |                  |

## СОДЕРЖАНИЕ

|                                                                                                                                                                                                    |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Т.М. Александриди, Ю.В. Рогачев<br><i>Автоматическая цифровая вычислительная машина М-1 (АЦВМ М-1)</i>                                                                                             | 10 |
| Т.М. Александриди, Ю.А. Лавренюк, Ю.В. Рогачев<br><i>ЭВМ М-2</i>                                                                                                                                   | 15 |
| Ю.В. Рогачев<br><i>ЭВМ М-3</i>                                                                                                                                                                     | 21 |
| З.Д. Алексеева, Я.А. Хетагуров<br><i>Алгоритмическая ЭВМ</i>                                                                                                                                       | 30 |
| Ю.А. Богоявленский<br><i>Подготовка специалистов по информационным и коммуникационным технологиям на базе семейства стандартов «Прикладная математика и информатика»</i>                           | 33 |
| Y.A. Bogoyavlenskiy<br><i>Education of Information-Communication Technology on the Base of Russian State Educational Standard «Applied Mathematics and Informatics»</i>                            | 45 |
| Н.П. Брусенцов, Рамиль Альварес Хосе<br><i>Троичные ЭВМ «Сетунь» и «Сетунь 70»</i>                                                                                                                 | 45 |
| Г.Б. Ефимов, Е.Ю. Зуева, И.Б. Щенков<br><i>Из истории развития и применения компьютерной алгебры в Институте прикладной математики имени М.В. Келдыша</i>                                          | 51 |
| В.Н. Захаров<br><i>Концепция новых поколений вычислительных систем – последняя крупная инициатива социалистических стран в области вычислительной техники и информатики – взгляд 20 лет спустя</i> | 52 |
| В.П. Климанов, М.В. Сутягин<br><i>Обеспечение надежности автоматизированных информационных систем на основе сетевой кластеризации серверов</i>                                                     | 53 |
| Л.Н. Королев<br><i>Влияние программирования на архитектуру вычислительных систем (от прошлого к будущему)</i>                                                                                      | 61 |
| В.С. Криворученко<br><i>Распределенные системы автоматизации аэродинамических исследований</i>                                                                                                     | 62 |

|                                                                                                                                                                                                     |            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| И.И. Левин                                                                                                                                                                                          |            |
| <b>История создания многопроцессорных систем с программируемой архитектурой</b>                                                                                                                     | <b>63</b>  |
| Н.А. Лукин                                                                                                                                                                                          |            |
| <b>Функционально-ориентированные процессоры для систем реального времени</b>                                                                                                                        | <b>63</b>  |
| Б.М. Малашевич                                                                                                                                                                                      |            |
| <b>Зеленоградский Центр микроэлектроники – история создания</b>                                                                                                                                     | <b>64</b>  |
| Л.Б. Малиновский                                                                                                                                                                                    |            |
| <b>Информационные технологии в Украине. Состояние и перспективы</b>                                                                                                                                 | <b>67</b>  |
| Ю.И. Митропольский                                                                                                                                                                                  |            |
| <b>Развитие архитектуры ЭВМ от БЭСМ-6 до суперкомпьютеров</b>                                                                                                                                       | <b>67</b>  |
| Ю.Н. Никольская                                                                                                                                                                                     |            |
| <b>Создание системы противоракетной обороны (ПРО) и СуперЭВМ»<br/>(по материалам доклада академика РАН В.С. Буричева)</b>                                                                           | <b>71</b>  |
| В. В. Пржиляковский, Ю.В. Карпилович, Г.Д. Смирнов                                                                                                                                                  |            |
| <b>Развитие промышленности средств вычислительной техники в Белорусской ССР</b>                                                                                                                     | <b>72</b>  |
| З.Л. Рабинович, Б.Н. Малиновский                                                                                                                                                                    |            |
| <b>О пионерских работах Киевской лаборатории С.А. Лебедева Института электротехники<br/>АН УССР и Института кибернетики им. В.М. Глушкова АН УССР<br/>в области создания и развития компьютеров</b> | <b>72</b>  |
| Ю.В. Рогачев                                                                                                                                                                                        |            |
| <b>Четыре поколения вычислительных машин М.А. Карцева (М-4, М4-2М, М-10, М-13)</b>                                                                                                                  | <b>82</b>  |
| Н.С. Рузанова                                                                                                                                                                                       |            |
| <b>Вычислительная техника в Карелии</b>                                                                                                                                                             | <b>96</b>  |
| Г.Г. Рябов                                                                                                                                                                                          |            |
| <b>Школа академика С.А. Лебедева. Первые шаги автоматизации проектирования ЭВМ</b>                                                                                                                  | <b>97</b>  |
| О.И. Семенов                                                                                                                                                                                        |            |
| <b>Очерк истории становления и развития вычислительной техники в Республике Беларусь</b>                                                                                                            | <b>97</b>  |
| М.Э. Смолевицкая                                                                                                                                                                                    |            |
| <b>Фондовое собрание «Электронные цифровые вычислительные машины» в Политехническом музее</b>                                                                                                       | <b>98</b>  |
| Ya.I. Fet                                                                                                                                                                                           |            |
| <b>Development of computer science in siberia</b>                                                                                                                                                   | <b>99</b>  |
| В.П. Шириков                                                                                                                                                                                        |            |
| <b>Распределенные системы обработки данных</b>                                                                                                                                                      | <b>105</b> |
| И.Р. Агамирзян                                                                                                                                                                                      |            |
| <b>С.С. Лавров и история программирования в СССР</b>                                                                                                                                                | <b>112</b> |
| O.L. Ahremchik                                                                                                                                                                                      |            |
| <b>Designing of the computing system for the automated control of production processes</b>                                                                                                          | <b>112</b> |
| М.Ш. Бадрутдинова                                                                                                                                                                                   |            |
| <b>Основные вехи развития вычислительной техники в стране через призму истории<br/>Казанского производственного объединения по выпуску вычислительных систем (КПО ВС)</b>                           | <b>115</b> |

|                                                                                                                                                 |            |
|-------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| С.Д. Белов, Л.Б. Чубаров, А.М. Федотов, В.С. Никульцев                                                                                          |            |
| <b>Развитие сети передачи данных СО РАН: итоги трех десятилетий</b>                                                                             | <b>120</b> |
| А.А. Берс                                                                                                                                       |            |
| <b>Электронная подготовка изданий</b>                                                                                                           | <b>120</b> |
| А.А. Берс                                                                                                                                       |            |
| <b>Рабочая станция «МРАМОР»</b>                                                                                                                 | <b>128</b> |
| Н.Ф. Богданова                                                                                                                                  |            |
| <b>История создания и деятельности Минского научно-учебного центра «Алгоритм»</b>                                                               | <b>134</b> |
| И.Ф. Богданова                                                                                                                                  |            |
| <b>Кандидатский минимум по информатике: история и современность (1987–2006)</b>                                                                 | <b>135</b> |
| Ю.А. Богоявленский, Д.Ж. Корзун, К.А. Кулаков, М.А. Крышень                                                                                     |            |
| <b>Проект Web-SynDic: Система удаленного решения линейных диофантовых уравнений в неотрицательных целых числах</b>                              | <b>136</b> |
| А.А. Бородин, Ю.А. Богоявленский                                                                                                                |            |
| <b>Полумарковская модель процесса захвата среды станцией в беспроводных сетях 802.11</b>                                                        | <b>145</b> |
| В.П. Боюн                                                                                                                                       |            |
| <b>Информационные основы и принципы построения интеллектуальных видеосистем</b>                                                                 | <b>152</b> |
| K.L. Bojanov                                                                                                                                    |            |
| <b>The development of Computer Science in Bulgaria and the cooperation of the Eastern European countries in the creation of computer series</b> | <b>152</b> |
| Н.П. Брусенцов, Ю.В. Владимирова                                                                                                                |            |
| <b>Троичная диалектическая информатика</b>                                                                                                      | <b>153</b> |
| М.А. Бульонков                                                                                                                                  |            |
| <b>Смешанные вычисления в Новосибирске</b>                                                                                                      | <b>159</b> |
| П.В. Воронина                                                                                                                                   |            |
| <b>Концепция представления академической организации в сети Интернет</b>                                                                        | <b>168</b> |
| А.И. Галушкин                                                                                                                                   |            |
| <b>Транспьютерные системы – начало становления в России ЭВМ с массовым параллелизмом</b>                                                        | <b>168</b> |
| <b>Индекс фамилий авторов статей</b>                                                                                                            | <b>188</b> |
| <b>Индекс наименований организаций</b>                                                                                                          | <b>189</b> |

Научное издание

# **SORUCOM.2006**

**РАЗВИТИЕ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ  
В РОССИИ И СТРАНАХ БЫВШЕГО СССР:  
ИСТОРИЯ И ПЕРСПЕКТИВЫ**

Материалы международной конференции

В двух частях

*Часть 1*

Редактор И.И. Куроптева. Компьютерная верстка – И.Г. Лежнев

Подписано в печать 19.06.06. Формат А4.

Бумага офсетная. Тираж 200 экз.

Государственное образовательное учреждение  
высшего профессионального образования  
ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
185910, г. Петрозаводск, пр. Ленина, 33