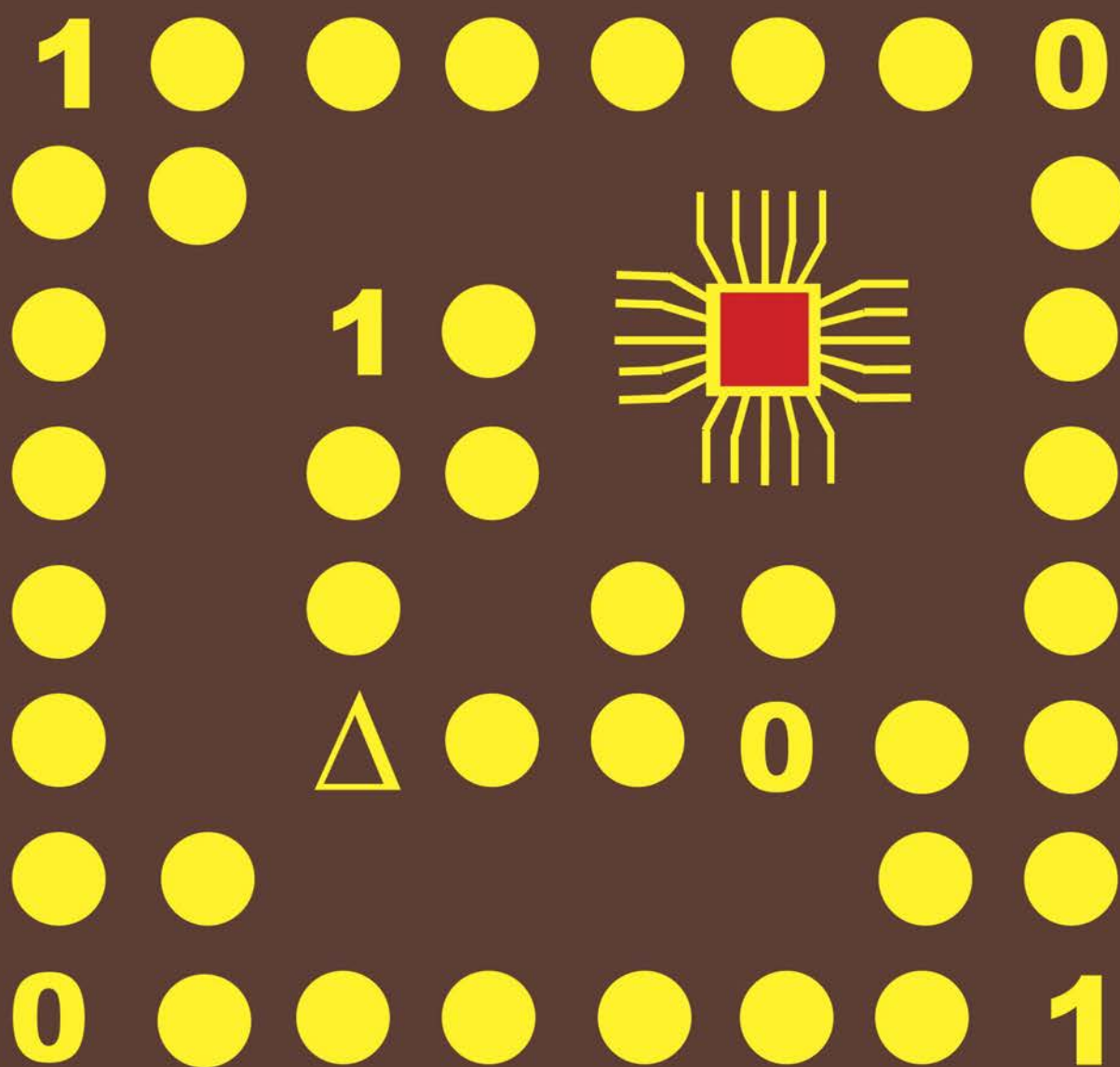


Международная научная конференция, посвященная 80-летию со дня рождения академика В. А. Мельникова

СБОРНИК ДОКЛАДОВ



МОСКВА, 19-20 февраля, 2009



(1928 – 1993)

Российская Академия наук
Институт системного программирования РАН
НИИ Системных исследований РАН
Научный Фонд «Первая исследовательская Лаборатория имени
академика В. А. Мельникова»

Международная
научная конференция,
посвященная 80-летию
со дня рождения
академика В. А. Мельникова

Сборник докладов

Выпускающие научные редакторы:

А. Н. Томилин (Институт системного программирования РАН)

М. Ш. Сургуладзе (НИИ системных исследований РАН)

Научный Фонд
«Первая исследовательская
Лаборатория имени академика
В. А. Мельникова»

г. Москва – 2009

Международная Научная конференция, посвященная 80-летию со дня рождения академика В. А. Мельникова – М. , 19-20 февраля 2009 г. : Сборник докладов
Некоммерческая Организация Научный Фонд «Первая Исследовательская Лаборатория имени академика В. А. Мельникова», 2009 г. – с. , ил.

Сборник докладов юбилейной конференции, посвященной 80-летию со дня рождения академика В. А. Мельникова, включает доклады ведущих отечественных и зарубежных ученых и специалистов по проблемам вычислительной техники и программирования, развития информационных технологий и вычислительных наук, совершенствования системной безопасности и информационно-вычислительных комплексов. Широкий спектр тематики представленных докладов отражает многогранность деятельности и научных интересов академика Владимира Андреевича Мельникова.

Пленарные доклады посвящены фундаментальным исследованиям в области вычислительной техники и программирования, построению перспективных вычислительных систем и их применению для решения важнейших научных задач.

Секционные доклады отражают актуальные вопросы реализации прикладных задач в различных областях науки, промышленности, обороны и т. п.

Лицензия ЛР №071872 от 24 мая 1999 г. Подписано в печать . Формат 60x90/8. Бумага офсетная. Гарнитура «Таймс».

НО Научный Фонд «Первая Исследовательская Лаборатория имени академика В. А. Мельникова»
РОССИЯ 117218 Москва, Нахимовский проспект, 36, корп. 1, комн. 816.
Тел. : +7(495) 718-2139, +7(965) 108-4477. E-mail: sfm12@mail. ru

Отпечатано с оригинал-макета в типографии.



- © Научный Фонд «Первая Исследовательская Лаборатория имени академика В. А. Мельникова», 2009
- © Оригинал-макет, оформление, обложка, предпечатная подготовка – Научный Фонд «Первая Исследовательская Лаборатория имени академика В. А. Мельникова», издательский отдел, 2009

Организаторы конференции

Российская Академия наук
Отделение математических наук РАН
Отделение нанотехнологий и информационных технологий РАН
НО Научный Фонд «Первая исследовательская Лаборатория имени академика В. А. Мельникова»

Организационный комитет конференции

Сопредседатели:

О. М. Белоцерковский академик РАН
В. Б. Бетелин академик РАН

Заместители:

А. Н. Томилин зав. Отделом ИСП РАН
М. Ш. Сургуладзе президент Научного фонда

Члены оргкомитета:

Н. А. Северцев зав. Отделом ВЦ РАН, вице-адмирал
А. Б. Жижченко зам. Академика-секретаря ОМН РАН
В. П. Иванников директор ИСП РАН
Б. М. Шабанов зам. Директора МСЦ РАН
С. А. Власов ученый секретарь ОНИТ РАН

Программный комитет

О. М. Белоцерковский	- академик	А. Б. Жижченко	- чл. -корр. РАН
К. А. Валиев	- академик	Л. Н. Королев	- чл. -корр. РАН
В. К. Левин	- академик	В. Н. Решетников	- д. ф. -м. н., проф.
Г. И. Савин	- академик	А. Г. Мадера	- д. т. н., проф.

Заседания конференции

Пленарное:..... 19 февраля 2009 г. , зал МИ РАН.
Прямая онлайн трансляция в Интернет

Научная сессия:..... 29 января 2009 г. , зал Президиума РАН

Секция 1:..... 25 февраля 2009 г. , зал ФТИ РАН

Секция 2:..... 20 февраля 2009 г. , зал ВЦ РАН

Секция 3:..... 27 февраля 2009 г. , зал НИИСИ РАН

Благодарности

**Организационный комитет конференции выражает
глубокую признательность:**

за финансовую поддержку конференции

Научному Фонду «Первая исследовательская Лаборатория имени академика В. А. Мельникова»
Российскому Фонду Фундаментальных Исследований

за сотрудничество, организационную и техническую поддержку конференции

Научному Фонду «Первая исследовательская Лаборатория имени академика В. А. Мельникова»
Институту системного программирования РАН
Отделению математических наук РАН
Отделению нанотехнологии и информационных технологий РАН
НИИ системных исследований РАН
Математическому институту имени В. А. Стеклова РАН
НИВЦ МГУ им. М. В. Ломоносова

**за помощь в проведении пленарного и секционных заседаний
и редактирования Сборника докладов конференции**

А. Д. Изааку	МИ им. В. А. Стеклова РАН
П. И. Кандалову	МИРЭА
А. И. Малинину	МСЦ РАН
Н. А. Ляликову	МСЦ РАН
Н. В. Никольскому	НИИСИ РАН
Н. А. Северцеву	ВЦ им. А. А. Дородницына РАН
М. Ш. Сургуладзе	НИИСИ РАН
Р. В. Кюрдиану	НИИСИ РАН
М. В. Михайлюку	НИИСИ РАН
А. Н. Томилину	ИСП РАН

за поддержку издания Сборника докладов

Научному Фонду «Первая исследовательская Лаборатория имени академика В. А. Мельникова»
Межведомственному суперкомпьютерному центру РАН
Электронной библиотеке «Научное наследие России»

Содержание

Пленарные доклады

Вступительное слово <i>Левин В. К.</i>	1
Системное программирование: состояние и тенденции <i>Иванников В. П.</i>	3
Суперкомпьютеры и дискретная топология <i>Рябов Г. Г.</i>	7
Развитие концепции специализации и мультиархитектуры в вычислительных системах <i>Митропольский Ю. И.</i>	10
Системы реального времени с самоконтролем вычислительных средств и схем на базовых БИС <i>Хетагуров Я. А.</i>	14
Новый класс сверточных кодов <i>Урелидзе Н. А.</i>	17
Московский энергетический институт и отечественные школы вычислительной техники <i>Топорков В. В.</i>	20
From Science to Business and back and fall of a Scientific Entrepreneurial Adventure <i>Dickhoven S. (Дикховен З.)</i>	31
Место кибернетики и информатики в общем (школьном) образовании <i>Семенов А. Л.</i>	35
Об идентификации нелинейных непрерывных систем <i>Шанишавили В. Г.</i>	39
Моделирование тепловых режимов в электронных модулях <i>Мадера А. Г., Кандалов П. И.</i>	42
Электричество и данные в одном потоке <i>Сургуладзе М. Ш.</i>	47
Вычислительные системы с программируемой структурой <i>Хорошевский В. Г.</i>	51
Влияние программирования на архитектуру вычислительных систем (от прошлого к будущему) <i>Королев Л. Н.</i>	63
Экспериментальные исследования с использованием отечественной вычислительной техники и методов некорректно поставленных задач <i>Горячев В. А.</i>	65

Секция 1. Информационные структуры и визуализация в тренажерно-обучающих системах

Руководители – Михайлюк М. В., Решетников В. Н. (НИИСИ РАН)

Общая структура программного обеспечения тренажерного комплекса <i>Михайлюк М. В., Торгашев М. А., Хураськин И. А.</i>	68
Физически корректное моделирование освещенности в задачах компьютерной графики <i>Волобой А. Г.</i>	72
Методика и алгоритмы визуализации для обучающих модулей компьютерных тренажерно-обучающих систем <i>Решетников В. Н., Мамросенко К. А.</i>	76
Архитектура ускорителя трехмерной графики для ЭВМ серии Багет <i>Евлампов Б. Е., Корниленко А. В., Костюхин К. А.</i>	80
Текстурирование больших регулярных сеток виртуальных сцен в имитационно-тренажерных комплексах <i>Тимохин П. Ю.</i>	84
Трехмерное динамическое моделирование работ по реабилитации пункта временного хранения ОЯТ и РАО <i>Степеннов Д. Б., Максимов С. Б.</i>	88
Система отслеживания движений человека на основе технологии беспроводных сенсорных сетей для визуализации аватара <i>Вабищевич А. Н., Панфилов П. Б.</i>	91
Опыт использования систем компьютерной графики в тренажерах ПКА <i>Масалкин А. И.</i>	96
Моделирование движения виртуальных объектов на основе скелетной анимации <i>Торгашев М. А.</i>	99
Особенности управления робототехническими системами космического назначения <i>Богомолов В. П.</i>	102
Моделирование и визуализация взаимодействия 3D-объектов сложной структуры в индуцированной виртуальной среде <i>Афанасьев В. О., Клименко С. В., Фомин С. К.</i>	106
Пространственно-временное моделирование и планирование промышленных проектов <i>Семенов В. А., Аничкин А. С., Казаков К. А., Тарлапан О. А.</i>	111
Коллективная инженерия на основе визуальных моделей <i>Семенов В. А., Морозов С. В., Энкович И. В.</i>	114

Современный взгляд на структуру систем автоматического построения трехмерных виртуальных моделей по изображениям <i>Юрин Д. В.</i>	118
Создание виртуального окружения в тренажерных комплексах <i>Хураськин И. А.</i>	122
Спутниковые сети связи стандарта DVB-RCS с многостанционным доступом MSF-TDMA <i>Генов А. А.</i>	126
SatExpress – пилотный проект двустороннего спутникового широкополосного доступа <i>Сургуладзе М. Ш.</i>	130
Секция 2. Материалы публикуются отдельной книгой «Фундаментальные проблемы системной безопасности» Руководитель – Северцев Н. А. (ВЦ им. А. А. Дородницына РАН)	133
Секция 3. Вычислительные науки: теория и практика Руководитель – Сургуладзе М. Ш. (НИИСИ РАН)	
Исследование подхода понижения потребляемой мощности нанотранзисторных цифровых устройств <i>Масальский Н. В.</i>	133
Статистический метод определения времени начала оптимизаций в динамическом оптимизирующем трансляторе <i>Гимпельсон В. Д.</i>	135
Реализация программного комплекса, моделирующего вычислительные комплексы с архитектурой SPARC V9 <i>Мешков А. Н.</i>	138
Разработка планировщика задач реального масштаба времени для ОС UNIX <i>Федотов Д. А.</i>	141
Построение многопроцессорного NUMA-сервера на базе процессорного модуля «Эльбрус-3S» <i>Зайцев А. И., Шерстнев А. Е.</i>	143
Новый тип наводок в МПП с локальной металлизацией в экранах <i>Тимофеев В. К.</i>	147
Открытие законов натурального ряда и перспективы их применения <i>Хренов В. П.</i>	155
Современные методы функциональной верификации цифровых проектов <i>Шабанов Б. М., Яицков А. С.</i>	160
Анализ оптического контроля соединений многослойных печатных плат <i>Козлов А. С.</i>	164
Краткий очерк развития САПР ТОПТРАН <i>Водолазский В. И.</i>	169
Методы и средства защиты программных продуктов САПР <i>Смирнов В. Ю.</i>	173
Системный анализ мониторинга экологической безопасности от воздействия автомобильного транспорта в мегаполисе <i>Неуен Куанг Тхыонг, Филлипова Н. В.</i>	176
Метод сетевого оператора для синтеза систем управления <i>Дивеев А. И.</i>	180
О безопасности функционирования динамической системы на основе технической устойчивости <i>Дружинина О. В., Северцев Н. А., д. т. н., Масина О. Н.</i>	184
Исследование управляемых динамических процессов методом предельных уравнений <i>Дружинина О. В., Климачкова Т. С.</i>	187
Об устойчивости и управляемости транспортных систем <i>Масина О. Н., Дружинина О. В., Иванова Ю. А., Игонина Е. В.</i>	190
Управление и стабилизация динамических систем на конечном интервале времени <i>Климачкова Т. С., Мулкиджан А. С.</i>	194
О природе турбулентности в конвекции Рэлея – Бенара (аннотация к статье) <i>Магницкий Н. А.</i>	198
Решение Международной научной конференции «Проблемы вычислительной техники и программирования», посвященной 80-летию со дня рождения академика В. А. Мельникова	199
Приложения (приветствия, телеграммы и поздравления участникам и т.п.)	200

Пленарные доклады

Вступительное слово

Левин В. К., академик РАН

Член бюро Отделения нанотехнологий и информационных технологий РАН

Конференция посвящена 80-летию со дня рождения выдающегося ученого и конструктора академика Владимира Андреевича Мельникова, чья научная деятельность была связана с созданием быстродействующих электронных вычислительных машин, комплексов и систем.

С начала 50-х годов в СССР создавались отечественные ЭВМ (практически одновременно с созданием ЭВМ за рубежом), необходимые для решения вычислительных задач важнейших государственных программ (ядерная физика, космические исследования, оборона). Для этого были созданы научные школы разработки компьютеров, вычислительной математики и программирования.

В научной школе академика С. А. Лебедева Владимир Андреевич вырос как ученый и конструктор. Он до конца своей жизни сохранил чувство глубокой любви и уважения к Учителю и вместе с соратниками и учениками успешно продолжил развитие его школы.

Трудовой путь В. А. Мельников начал с 1950 года студентом Московского энергетического института (окончил его по специальности «Автоматика и телемеханика»), а затем сотрудником Первой лаборатории Института точной механики и вычислительной техники АН СССР, активно участвуя в создании одной из первых в СССР ЭВМ – БЭСМ-1. Запущенная в эксплуатацию в 1953 году БЭСМ-1 имела 5 тысяч электронных ламп, быстродействие 8-10 тысяч операций в секунду и являлась самой быстродействующей машиной в Европе и одной из самых быстродействующих машин в мире. В. А. Мельников разрабатывал устройство управления машины. Принципы организации и конструкции БЭСМ-1 воплотились и совершенствовались в последующих ЭВМ, разработанных в СССР.

Еще молодым инженером В. А. Мельников становится руководителем следующей разработки института – БЭСМ-2 и ее наладки. В создании этой и последующих машин серии «БЭСМ», боль-

ших вычислительных комплексов и суперЭВМ проявились лучшие качества Владимира Андреевича – увлеченность, самоотверженность в работе, умение найти и реализовать комплексные инженерно-конструкторские решения. Созданные В. А. Мельниковым вместе с возглавляемым им коллективом надежные вычислительные машины и системы заслужили высокий авторитет и явились в нашей стране основным средством решения важнейших научных и технических задач и оперативного управления работой больших систем.

Целой эпохой в отечественной вычислительной технике было создание ЭВМ БЭСМ-6, обеспечившей решение важнейших вычислительных задач в научных исследованиях по ядерной физике (в Москве, Арзамасе, Челябинске, Дубне и других научных центрах), а также явившейся основным инструментом для создания крупных систем управления и базой развития информатики в нашей стране. В. А. Мельников, как заместитель главного конструктора академика С. А. Лебедева, возглавил коллектив разработчиков, который за короткий срок завершил весь комплекс работ по созданию элементной базы, конструкции, программного обеспечения.

Следующий этап деятельности В. А. Мельникова был связан с созданием системы обработки данных АС-6, воплотившей в себе многие идеи, составившие основу будущих суперЭВМ. Она использовалась совместно с БЭСМ-6 в космической программе «Союз – Аполлон» и последующих запусках космических кораблей. Большой вклад В. А. Мельников внес и в создание на базе АС-6 – БЭСМ-6 информационно-вычислительных комплексов центров управления полетами космических аппаратов.

В последнее десятилетие жизни, будучи директором Института проблем кибернетики РАН и возглавляя также коллектив разработчиков в Министерстве электронной промышленности СССР, Владимир Андреевич все свои силы отдавал выполнению работ по созданию первой отечествен-

ной суперЭВМ «Электроника СС БИС». Он сумел объединить для этого творческий потенциал и производственные возможности институтов Академии наук и предприятий промышленности.

Академик В. А. Мельников уделял много сил образованию и постоянно заботился о подготовке молодых специалистов. Он руководил работой кафедр в МФТИ, МИРЭА и МГУ, принимал активное участие во внедрении информатики и вычислительной техники в школьное образование, являясь главным редактором журнала «Информатика и образование». Академик В. А. Мельников считал, что важнейшее для страны дело – развитие образования всех уровней – в современных условиях нуждается в поддержке информационных технологий.

В. А. Мельников вел большую научно-организационную работу, являясь заместителем академика-секретаря Отделения информатики, вычислительной техники и автоматизации АН СССР, членом Президиума Высшей аттестационной комиссии при Совете Министров СССР, главным редактором сборника «Кибернетика и вычислительная техника». Он также вел большую работу по организации международных научных связей Академии наук СССР в области информатики и вычислительной техники, являясь ответственным за соответствующие направления сотрудничества с Академиями наук социалистических стран и научными организациями ряда капиталистических стран.

Большая и плодотворная работа академика В. А. Мельникова отмечена высокими наградами родины – орденом Ленина, двумя орденами Трудового Красного знамени, медалями. В. А. Мельников – лауреат Государственных премий СССР (1969 и 1980 г. г.), лауреат Премии имени С. А. Лебедева Президиума АН Украины.

В. А. Мельников был избран членом-корреспондентом, а затем и действительным членом Академии наук СССР по Отделению математики, что свидетельствует о высокой оценке вклада ведущего разработчика компьютеров в обеспечение развития вычислительной математики, глубокой связи математики и информатики.

Представленные на научной конференции, посвященной памяти академика В. А. Мельникова пленарные доклады, а также доклады, сделанные на трех научных секциях, посвящены актуальным проблемам вычислительной техники, информатики и применению ЭВМ в математическом моделировании и научных исследованиях.

Системное программирование: состояние и тенденции

Иванников В. П., академик РАН

Директор ИСП РАН

Технологии разработки ПО

Рост размеров кода ОС

Система	Год	Размер (106 строк - LOC)
Windows 3. 1	1992	3
Windows NT 3. 5	1994	10
Windows NT 4. 0	1996	16
Windows 2000	1999	30
Windows XP	2001	45
Linux Kernel 2. 6. 0	2003	5,7
Open Solaris	2005	9,7
Mac OS X 10. 4	2006	86
Windows Vista	2007	> 50

Другие показатели развития

- Эскалация размеров и сложности
- Увеличение функциональных возможностей
- Рост объемов перерабатываемых данных
- Расширение использования параллелизма и распределенности
- Рост требований к переносимости и совместимости

Текущие достижения

- Объектно-ориентированные методы анализа и проектирования.
- Компонентные технологии
- Стандартизация и уточнение семантики языков программирования и библиотечных интерфейсов
- Широкое использование итеративных процессов разработки
- Расширение области применения формальных методов (model checking и др.)

Проблемы

- Количество ошибок на 1000 строк неоттестированного кода остается неизменным

Программные системы	Число ошибок на 1000 строк кода
Среднее по индустрии (McConnell)	15-50
Microsoft (до тестирования)	10-20
NASA JPL (до тестирования)	6-9
Linux	~7
Microsoft (продукты)	0.5
NASA JPL (продукты)	0.003

- Формальные методы разработки и анализа применимы лишь к небольшим системам

Технологии разработки ПО

Новые вызовы

- Распределенные адаптивные и динамически перестраивающиеся системы
- Масштабируемые технологии разработки и анализа свойств ПО на базе формальных методов
 - *Интеграция различных методов верификации: дедуктивный анализ, проверка моделей, тестирование на основе формальных моделей, статический анализ*
- Технологии разработки распределенных систем с заданными характеристиками по многим показателям качества
 - *Выполнение разнородных требований: корректность, надежность, защищенность, производительность, удобство использования и развития*
- Формализация стандартов на программные интерфейсы разных уровней

Анализ программ

Текущие достижения

- Глобальная и межпроцедурная оптимизация, межпроцедурный анализ указателей.
- Исследование гнезд циклов на распараллеливаемость: Омега-тест, изменение порядка циклов в гнезде.
- Машинно-ориентированная оптимизация: распределение регистров, планирование кода.
- Языки нового поколения (Java, C#). Совмещение интерпретации и динамической компиляции (JIT), адаптивная оптимизация.
- Low Level Virtual Machine (LLVM) и его интерпретатор (Apple): адаптивная оптимизация для C и C++.
- Методы анализа успешно применяются для решения проблем в смежных областях: обратная инженерия, поиск уязвимостей безопасности.

Проблемы

- Разрыв между возможностями аппаратуры (параллелизм на всех уровнях) и возможностями анализа.
- Расширение области применения специализированных платформ: Cell, GPU, FPGA.
- Низкая «продуктивность» разработки приложений для параллельных систем:
 - *отсутствие языков высокого уровня и соответствующих технологий (HPF, Cilk (MIT), UPC (Berkeley) не привели к успеху);*
 - *программирование с использованием MPI (распределенная память) и OpenMP (общая память).*
- Отсутствие стандартов и устоявшихся технологий для специализированных платформ.

Тенденции решения проблем

- Исследование и моделирование различных классов приложений (dwarf - Berkeley) и аппаратных платформ с целью разработки адекватных средств (языки, инструменты), обеспечивающих эффективное отображение приложения на конкретную аппаратную платформу.

- Новые языки параллельного программирования:
X10 (IBM), Chapel (Cray), Fortress (Sun).
- Технологии и соответствующие среды поддерживающие разработку на уровне MPI/OpenMP
- Технологии программирования для специализированных платформ:
 - *низкоуровневые: CUDA (NVidia), CAL (AMD);*
 - *языки высокого уровня: Brook+ (AMD), Ct (Intel).*

Управление данными

Текущие достижения

- СУБД (Oracle, IBM, Microsoft и т. д.)
 - *эффективная и масштабируемая обработка баз данных объемом в сотни терабайт и даже петабайт;*
 - *развитые средства оптимизации запросов на основе оценочных методов;*
 - *встроенные в серверы средства OLAP и data mining;*
 - *полнотекстовый поиск и поддержка XML;*
 - *параллельные серверы.*
- Управление данными в Internet
 - *развитые средства полнотекстового поиска по ключевым словам;*
 - *собственные средства управления данными, хорошо распараллеливающиеся на основе механизма map-reduce (Google, Yahoo!).*

Проблемы

- Сложность администрирования, настройки и использования СУБД.
 - *возрастающий дефицит специалистов;*
 - *невозможность полноценного использования SQL.*
- Сложность разработки приложений баз данных.
 - *Проблема «потери соответствия (impedance mismatch) между языками программирования и системами баз данных;*
 - *невозможность использования семантического поиска;*
 - *незрелость средств управления данными.*
- Неполное использование новых возможностей аппаратуры.
 - *многоядерные и многопоточковые архитектуры, основная память большого объема, флэш-память и т. д.*

Тенденции решения проблем

- Переход от универсальных к специализированным системам управления данными
 - *XML-СУБД, системы с хранением данных «по столбцам» и т. д.*
- Разработка новых языков программирования, освобождающих разработчиков от проблемы «impedance mismatch»
 - *Ruby-on-Rails, LINQ*
- Повышение уровня семантики поиска в Internet
 - *использование массивных структурированных корпусов текстов для автоматического построения онтологий*

Clouds Computing

- Публично доступные службы для запуска программ.
- Позволяют пользоваться параллельной аппаратурой для решения, в частности, задач, связанных с обработкой больших объемов данных.
- Имеющиеся параллельные СУБД не могут масштабироваться до тысяч параллельных узлов.
- Необходимо понять, как и для чего разумно использовать «облачные вычисления» при управлении данными и разработать соответствующие программные системы .

Проблемы компьютерной безопасности

Обеспечение компьютерной безопасности – одна из наиболее актуальных задач системного программирования.

Рынок ПО для обеспечения защиты информации в последние годы растет гораздо быстрее, чем весь рынок ИТ.

Причины:

1. Глобализация информационного пространства (широкое распространение сетевых технологий, интенсивное развитие средств телекоммуникации);
2. Внедрение электронных средств хранения и обработки информации во все сферы экономической, политической и военной деятельности (документооборот, связь, базы данных, системы поддержки принятия решений и др.);
3. Ускорение и удешевление разработки ПО в ущерб его качеству в условиях острой конкуренции на рынке ИТ;
4. Бытовая компьютеризация (средства мобильной связи, беспроводные сети и пр.)

Угрозы

1. Выведение из строя ПО (распространение вредоносных программ, компьютерных «вирусов», троянских программ, подмена мобильного кода).
2. Выведение из строя компьютерных сетей (создание перегрузок в сети, сетевые вторжения)
3. Создание и распространение ПО со скрытыми дефектами (внедрение в ПО закладок, встроенных уязвимостей и пр.).
4. Нарушение конфиденциальности и целостности информации (несанкционированный доступ к базам данных и электронным архивам, распространение программ-шпионов, взлом криптосистем).
5. Компьютерное пиратство и нарушение авторских прав на ПО (нелегальное копирование и распространение ПО).

Средства обеспечения

1. Антивирусные программы, интеллектуальные системы мониторинга, системы защиты от утечки данных
2. Системы обнаружения и предотвращения сетевых вторжений (сетевой мониторинг, межсетевые экраны, шлюзы, брандмауэры).
3. Использование ПО с открытым кодом (Linux vs Microsoft).
4. Сертификация и аудит ПО, системы верификации и поиска уязвимостей в программах.
5. Политики безопасности в информационных системах (разграничение доступа, аутентификация и авторизация).
6. Криптографические средства защиты информации (системы шифрования, электронной подписи, криптографические протоколы, обфускация программ и пр.).
7. Отслеживание распространения ПО (водяные знаки, отпечатки пальцев).

Направления исследований

Теоретические исследования:

- новые математические модели информационных систем, формализующие ключевые понятия компьютерной безопасности (вторжение, аномальное поведение, стойкая защита и др.),
- методы и алгоритмы информационной защиты ПО, оценки их эффективности и стойкости.

Прикладные разработки:

- инструментальные средства проверки безопасности ПО,
- открытые ОС со встроенным механизмом обеспечения информационной безопасности,
- компьютерные сети, ориентированные на обеспечение коллективной безопасности абонентов.

Суперкомпьютер и дискретная топология

Рябов Г. Г., *член-корр. РАН*

(НИВЦ МГУ им. М. В. Ломоносова)

Для моделирования объектов дискретной топологии привлекаются суперкомпьютерные системы, нередко из списка TOP-500, что вполне объяснимо комбинаторным характером большинства задач, особенно при исследовании динамики перестроек таких структур [2,4]. В статье рассмотрены три метода кодирования, которые имеют перспективы развития и связаны с распараллеливанием вычислений в рассматриваемой области: **1.** Кодирование кубических комплексов. **2.** Кодирование для вычисления переходных вероятностей в цепях Маркова при перестройках триангуляций евклидовых пространств. **3.** Кодирование топологических ситуаций при преобразованиях комплексов – дискретных аналогов гомотопных преобразований. База, выбранная для геометрико-топологических объектов, обладает «минимальными» математическими ресурсами (целые точки и примитивные вектора в евклидовых пространствах).

Рассматриваются евклидовы пространства \mathbf{R}^n и в них подпространства целых точек \mathbf{Z}^n , как вершин объектов (граней, симплексов, комплексов и т. д.) В качестве множества ребер (граней размерности 1) используются отрезки-ребра, инцидентные вершинам в целых точках и коллинеарные примитивным (простым) векторам с целочисленными координатами и не имеющими внутренних целых точек. Если координаты примитивных векторов не превосходят по модулю целого p , то будем обозначать такое множество ребер V_p . Поэтому исходную базу обозначим $\{\mathbf{Z}^n, V_p\}$. В данной статье основные построения будут относиться к случаю $\{\mathbf{Z}^n, V_1\}$.

Кодирование кубических комплексов

Вначале рассматриваются все кубические комплексы в рамках n -мерного единичного куба \mathbf{I}^n . Под кубическими комплексами здесь подразумеваются комплексы из k -граней (граней размерности k ; $0 \leq k \leq n$), поскольку каждая грань в \mathbf{I}^n есть «куб» соответствующей размерности. В

[7] на основе анализа свойств коэффициентов пирамиды Паскаля (обобщение треугольника Паскаля для трехмерного случая) установлено свойство **биекции** между множеством всех n -разрядных троичных слов и множеством всех k -граней в \mathbf{I}^n .

Пусть в \mathbf{R}^n задан репер $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n$ и задано $D = d_1, d_2, \dots, d_n$ – n -разрядное троичное слово; d_i из алфавита $\{0; 1; 2\}$. При этом установлено од-

нозначное соответствие между номером разряда в слове и номером реперного вектора $\mathbf{e}_i \rightarrow d_i$.

Выберем некоторую k -грань и положим в D все $d_i = 2$, если в выбранной грани содержится единичное ребро, коллинеарное \mathbf{e}_i . Так в D будет k «двоек», а остальные разряды будут иметь значения 0 и 1 и определять «место» этой грани в \mathbf{I}^n , т. е. задавать параллельный перенос в это «место» (трансляцию).

Определим $\chi_2(D)$ (характеристику размерности) как n -разрядное двоичное слово, разряды q_i которого удовлетворяют соотношению:

$$q_i = \begin{cases} 1, & \text{если } d_i = 2; \\ 0, & \text{в противном случае;} \end{cases}$$

При таком определении $\sum q_i = k$, где k – размерность грани в \mathbf{I}^n .

Определим $\chi_1(D)$ (характеристику трансляции, как n -разрядное двоичное слово, разряды t_i которого удовлетворяют соотношению:

$$t_i = \begin{cases} 0, & \text{если } d_i = 2; \\ d_i, & \text{в противном случае;} \end{cases}$$

Формально можно записать, что грань удовлетворяет соотношению:

$$f(D) = \prod q_i \mathbf{e}_i + \sum t_i \mathbf{e}_i,$$

где Π – перемножение множеств, а $q_i \mathbf{e}_i = \mathbf{e}_i$ при $q_i = 1$ и отсутствие \mathbf{e}_i при $q_i = 0$. Аналогично для $t_i \mathbf{e}_i$.

Грубо говоря, каждая k -грань «формируется» из набора k реперных ребер около точки $(0,0,\dots,0)$, а затем транслируется (параллельно сдвигается) в Γ^n .

Так троичное слово 202211 представляет в Γ^6 трехмерную грань (3-куб) с ребрами, коллинеарными e_1, e_3, e_4 и транслированную на вектор (000011).

Поставим в соответствие комплексу в Γ^n из m граней строку из m троичных n -разрядных слов. Так строка 202201;022102; 221200;202101 представляет комплекс в Γ^6 из трех 3-граней и одной 2-грани. Этот комплекс связный и имеет общую вершину (001101). Это устанавливается поразрядными операциями, определяющими пересечения граней и комплексов и детальное описание которых здесь не рассматривается.

Таким образом формально общее число всех возможных кубических комплексов в Γ^n равно мощности множества всех подмножеств граней всех размерностей, т. е. 2^A , где $A=3^n$. Реально их число значительно меньше, поскольку формальная строка может содержать слова, относящиеся к граням, где одна грань содержится в другой. Определение таких поглощений, как и пересечений граней и комплексов также сводится к поразрядным операциям.

В \mathbf{R}^n кубический комплекс в общем случае состоит из кубических комплексов разных Γ^n , привязанным к координатам (x_1, x_2, \dots, x_n) целых точек. Такая привязка возможна двумя способами.

1-ый способ. Снабдить каждую строку-комплекс из Γ^n n -мерной координатой этого Γ^n в \mathbf{R}^n .

2-ой способ. В n -мерном клеточном массиве в памяти компьютера в каждой клетке поместить строку соответствующего комплекса из Γ^n .

Во многих случаях, когда перестройка комплексов в каждом Γ^n зависит от ситуации в комплексах соседних координат, 2-ой способ имеет конструктивные преимущества.

Такой способ сродни постройке панельного дома с огромным разнообразием топологических «планировок» помещений.

В заключение этого раздела отметим особенность троичного слова (кода грани), которую можно характеризовать как «полуквантовость». Разряды, которые содержат 2, можно рассматривать как q -бит (из терминологии квантовых вычислений), поскольку за этой двойкой стоят и 1 и 0 и весь отрезок вещественных чисел $[0;1]$ соответствующего ребра из Γ^n . Разряды со значениями 0 и 1 являются как бы обычными двоичными. Поэтому условно каждое слово-

грань=квантовая составляющая (размерность) плюс двоичная составляющая (трансляция).

Дальнейший путь изучения динамики перестроек комплексов (строк троичных слов) связан с введением действий слева симметрической группы S_n или ее подгрупп, на строку как множество троичных слов, и ведет в область алгебраической топологии.

Кодирование полного множества элементарных событий и корректное вычисление переходных вероятностей в Марковских цепях

В [8] рассматриваются примитивные триангуляции \mathbf{R}^3 и \mathbf{R}^4 также на базе $\{\mathbf{Z}^n, V_1\}$, т. е. триангуляции, которые можно задать различными вариантами расположения диагоналей в гранях куба. Вводятся вектора диагональных степеней вершин, полностью описывающие тип триангуляции куба и инвариантные относительно вращений и отражений. Для каждого такого вектора рассматриваются все возможные варианты перехода его в другой при изменении направления диагонали в одной или нескольких гранях. Для этого строится отображение триангуляции куба в двоичный код для заданного изначально порядка следования граней в развертке куба (плоской для 3d и трехмерной для 4d). «0» в i -ом разряде 6-разрядного (по числу граней) кода для \mathbf{R}^3 соответствует диагонали, проведенной в i -ой грани из левого нижнего угла в правый верхний, а «1»-диагонали из правого нижнего угла.

Каждое элементарное событие (перестройка триангуляции) есть отображение одного кода развертки в другой. На всем множестве пар кодов $2^6 \times 2^6$ (для \mathbf{R}^3) мы определяем корректно, в смысле аксиоматики Колмогорова, переходные вероятности для Марковских цепей, описывающих случайный процесс диагональных перестроек. Эргодические и периодические свойства этих цепей полностью определяются матрицами переходных вероятностей этих процессов. С помощью такого метода кодирования получены стационарные распределения типов триангуляции в \mathbf{R}^3 при случайных перестройках одной, двух, трех, ... шести диагоналей в кубе за единицу дискретного времени.

Кроме того исследовалось распределение всех неконгруэнтных вершинных полиэдров при такой триангуляции. Вершинный полиэдр в \mathbf{R}^3 образуется из вершинного комплекса при примыкании симплексов из триангулированных кубов-октантов. Такое распределение отнесено к кодировке типов полиэдров в структуре Бозе-Эйнштейна. Различные типы (их 5) триангуляции

\mathbf{R}^3 рассматриваются как энергетические уровни, и частица, соответствующая типу триангуляции каждого из 8 октантов (вокруг вершины) помещается на соответствующий уровень. Таким образом имеем размещение 8 частиц в 5 ящиках (их 495) и для каждого такого «квантового» кода вычислено число неконгруэнтных вершинных полиэдров. Установлено, что в \mathbf{R}^3 для пяти таких «квантовых» кодов не существует ни одного типа полиэдра, т. е. спектр разрывен.

Кодирование топологических ситуаций при дискретных аналогах гомотопных преобразований

В [1,3,5] предложены методы полиэдризации (разбиения) пространства на полиэдры-зонотроны (одного типа) – дискретный аналог окрестности на базе $\{Z^n, V_1\}$. На таком построении определяются локальные преобразования расширения и сжатия без склеек и разрывов. Множество таких локальных преобразований и есть дискретный аналог гомотопных преобразований. Самым глубоким циклом при реализации расширения или сжатия является процедура проверки отсутствия разрыва или склейки, т. е. корректности каждого локального топологического действия. Такая процедура сводится к определению связных компонент на вершинном полиэдре. При однородной полиэдризации \mathbf{R}^3 с помощью кубододекаэдра с числом вершин 14 и числом ребер 50 во-первых надо хранить лишь один эталон полиэдра (ребра виртуальны), а во-вторых анализ топологической корректности (примерно эквивалентен 700 операций) заменяется табличным хранением исходов всех вариантов, которые заранее рассчитаны т. е. 2^{14} битов (затраты по памяти -мегабайт). Это дает сокращение машинного времени примерно в 1000 раз для \mathbf{R}^3 , а для \mathbf{R}^4 при числе вершин 30 и числе ребер более 10000 раз (расходы по памяти - Гбайт).

В определенном смысле идет размен скорости на емкость памяти. При такой организации процесс гомотопного расширения множества до объема решетки $10^9(1000 \times 1000 \times 1000)$ вершин и числа ребер 10^{13} на суперкомпьютере МГУ «Чебышев» (60 терафлопс) идет несколько минут.

Заключение

Приведенные примеры взяты как фрагменты методов, используемых при создании в НИВЦ МГУ инструментального комплекса [6] для операций с объектами дискретной топологии (общее название «топологический процессор»), ориентированного на современные суперкомпьютеры массивно-параллельной архитектуры. Примеры показывают дополнительные возможности эффективного использования суперкомпьютеров при подходе, который можно было бы характеризовать по аналогии с геометрией чисел, как «топология кодов».

Такое кодирование допускает во многих случаях поразрядные операции, что близко к предельным схемам распараллеливания вычислений до разряда, что особенно важно при высоких размерностях пространств.

Предложенные методы могут найти свое аппаратное воплощение в виде сопроцессоров к суперкомпьютеру.

Литература

1. Couprie M. , Bertrand G. Simplicity surfaces: a new definition of surfaces in Z^3 . SPIE Proceedings, vol. 3454, 1998
2. Latapy M. Generalized integer partitions, tilings of zonotopes and lattices. arXiv:math/0008022v2 [math. CO] 2000
3. Kenmochi Y. , Imiya A. Discrete polyhedrization of lattice point set. LNCS Proceedings. Vol. 2243, 2001
4. Desoutter V. , Destainville N. Flip dynamics in three- dimensional random tilings. arXiv:cond-mat/0406728v3[cond-mat. stat-mech] 2004
5. Рябов Г. Г. Метрические и топологические волны на решетках. Изд. МГУ. 2005
6. Ryabov G. , Serov V. , Simplicial-lattice model and metric-topological constructions. PRIP Proceedings, vol 2, 2007
7. Рябов Г. Г. О путевом кодировании k-граней в n-кубе. /Вычислительные методы и программирование. МГУ т. 9 N1,2008.
8. Рябов Г. Г. Цепи Маркова в динамике примитивной триангуляции R^3 и R^4 . / Вычислительные методы и программирование. МГУ. т. 10 N1,2009

Развитие концепции специализации, неоднородности и мультиархитектуры в вычислительных системах

Митропольский Ю. И. , *член-корреспондент РАН*

Физико-технологический институт РАН

Рассмотрено развитие концепции специализации аппаратных средств в ЭВМ БЭСМ-6, системе обработки данных АС-6 и суперкомпьютерной системе «Электроника СС БИС». На последующих этапах была развита концепция построения мультиархитектурных (неоднородных) вычислительных суперсистем. Разработан проект мультиархитектурной масштабируемой суперсистемы, производительность которой при использовании соответствующей элементной базы может превысить 1 PFLOPS. Благодаря согласованию форм параллелизма задач с формами параллелизма аппаратных средств достигается высокая степень эффективности системы. Архитектура масштабируемых процессоров обеспечивает снижение потерь производительности на аппаратном уровне. В настоящее время ведущие суперкомпьютерные фирмы мира проводят исследования и разработки в том же направлении и концептуально повторяют результаты наших исследований.

Вычислительная машина БЭСМ-6, Главным конструктором которой был академик С. А. Лебедев, а его заместителем академик В. А. Мельников, предназначалась для решения крупных научно-технических задач, что, естественно, отразилось как на ее архитектуре, так и на выборе системы элементов и конструкции. Система команд и аппаратура были ориентированы на эффективное выполнение операций с плавающей запятой, сокращение потерь при обращении к памяти и высокой пропускной способности каналов обмена с внешней памятью. 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. Для согласования пропускных способностей процессора и оперативной памяти применялось расслоение оперативной памяти и

неадресуемая буферная память с ассоциативным поиском.

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 и набор медленных направлений, аппаратура для которых ограничивалась минимальным специализированным набором согласующих элементов и схем связи этих элементов с процессором. Функционирование медленных направлений обеспечивалось программами работы с каждым конкретным типом устройства [1].

Таким образом, эффективность аппаратуры и высокая производительность при решении задач были обеспечены за счет специализации как схемотехнических, так и архитектурных решений.

Следующей крупной разработкой, выполненной под руководством Главного конструктора академика 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. 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. В системе АС-6 были впервые реализованы новые идеи, явившиеся основой разработок суперкомпьютеров и фундаментальных исследований по архитектуре перспективных вычислительных систем. АС-6 – неоднородная многомашинная вычислительная система. Проблемная ориентация центрального процессора АС-6 направлена на решение задач по управлению сложными объектами и эффективную трансляцию. Функциональная специализация периферийной машины ПМ-6, различных контроллеров и других вспомогательных устройств и специализация внутрисистемных каналов обеспечили высокую эффективность.

Разработка суперкомпьютерной системы «Электроника СС БИС-1», осуществленная под руководством академика В. А. Мельникова, 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.

Достижение максимальной производительности было возможно только при использовании синхронных конвейерных схем, более перспективной была признана векторно-конвейерная архитектура. Пиковая производительность двух-процессорной системы составила 500 MFLOPS [3]. 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. Программное обеспечение состояло из операционных систем основной и внешних машин, систем программирования на языках макроасемблера, «Фортран 77», «Паскаль», «Си».

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.

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 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 необходима многопроцессорная подсистема.

На первом этапе были разработаны концепции построения неоднородных суперсистем, в частности объединение в едином вычислительном комплексе векторного унипроцессора с мультипроцессором на микропроцессорах на основе доступа к общей памяти [4]. В систему могут входить несколько таких комплексов, 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 [7]. Суперсистема состоит из вычислительной подсистемы, мониторно-моделирующей подсистемы, системной и внешней памяти. Вычислительная подсистема имеет масштабируемую структуру и состоит из набора мультиархитектурных вычислительных комплексов. Каждый комплекс содержит набор мультипроцессоров и мониторную машину, тесно связанных между собой. В этот набор могут входить векторный и скалярный мультипроцессоры, состоящие соответственно только из векторных и скалярных процессоров, и мультиархитектурный мультипроцессор, состоящий из векторных и скалярных процессоров. Разработан набор модулей для построения процессоров разных типов.

Основным назначением векторного мультипроцессора является выполнение программ с явным параллелизмом на уровне данных. Такие программы должны быть векторизованы и ориентированы на выполнение сложных векторных операций. Архитектура, методология программирования, структура аппаратных средств, схемотехника и конструкция ориентированы на достижение максимальной производительности при выполнении таких программ.

Мониторно-моделирующая подсистема выполняет анализ программ (в том числе и в интерактивном режиме) с целью выявления их форм параллелизма, подготовки и распределения заданий между составными частями вычислительной

подсистемы. Она также управляет доступом к внешней полупроводниковой памяти, к дисковой и архивной памяти и к средствам отображения информации. В эту подсистему входит моделирующая машина и мониторные машины вычислительных комплексов. Все эти машины могут быть реализованы на основе серийных серверов, однако специфика задач по анализу программ и интерактивному взаимодействию с программистами может потребовать создания специализированных аппаратно-программных средств.

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, 9].

В 2006 г. фирма Cray объявила о новой стратегии для следующего поколения суперкомпьютеров, основанной на концепции неоднородных вычислительных систем. Новые системы фирмы используют принцип адаптивных вычислений на суперкомпьютерах за счет объединения различных по архитектуре систем на единой платформе [10]. В ноябре 2007 года фирма Cray объявила о выпуске семейства суперкомпьютеров Cray XT5 и XT5h, в состав которых входят скалярные, векторные и реконфигурируемые узлы [11].

Фирмы IBM, Sony и Toshiba совместно разработали неоднородную многопроцессорную систему на одном кристалле в проекте Cell [12]. В мае 2008 г. фирма IBM объявила о завершении тестирования суперкомпьютера Roadrunner, построенного на кристаллах Opteron фирмы AMD и кристаллах Cell, изготовленных по технологии 65 нм. Этот суперкомпьютер стал первым компьютером, преодолевшим рубеж 1 PFLOPS [13].

В Японии объявлено о проекте создания мультисистемы с производительностью 10 PFLOPS, состоящей из скалярных и векторных процессоров. К работе привлечены фирмы Hitachi, NEC и Fujitsu [14].

10 ноября 2008 г. было объявлено, что суперкомпьютер Jaguar фирмы Cray достиг производительности при решении реальной задачи 1,3 PFLOPS [15].

Реализация результатов описанных исследований может стать одним из катализаторов развития отечественной микроэлектроники и вычислительной техники.

Литература

1. Митропольский Ю. И. , БЭСМ-6, АС-6 и их влияние на развитие отечественной вычислительной техники // Информационные технологии и вычислительные системы, ОИТВС РАН, ИМВС РАН, 2002, № 3, С. 49 – 58.
2. Мельников В. А. , Митропольский Ю. И. , Малинин А. И. , Романков В. М. , Требования к конструкции высокопроизводительных ЭВМ и проблемы ее реализации // Сборник «Вопросы кибернетики», «Комплексное проектирование элементарно-конструкторской базы суперЭВМ», под ред. В. А. Мельникова и Ю. И. Митропольского, НСК АН СССР, 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, ОИВТА РАН, М. – 1993. - С. 28 – 41.
4. Митропольский Ю. И. , Концепции построения неоднородных вычислительных суперсистем // Сборник «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 // Сборник «Распределенная обработка информации». Труды Пятого Международного семинара, Новосибирск. - 1995. - С. 22 – 27.
6. Митропольский Ю. И. Архитектура мультиконвейерного модульного масштабируемого унипроцессора // Труды Шестого Международного семинара Распределенная обработка информации. 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». Москва. МГУ. - 2003. - С. 131-136.
8. Митропольский Ю. И. Мультиархитектура – новая парадигма для суперкомпьютеров // Электроника: наука, технология, бизнес. - 2005. - № 3. - С. 42 – 47.
9. Митропольский Ю. И. Проект многоуровневой масштабируемой мультиархитектурной вычислительной системы // Труды Четвертой международной конференции «Параллельные вычисления и задачи управления», Москва, 27-29 октября 2008 г. , Институт проблем управления им. В. А. Трапезникова, 2008, С. 533 – 558. PACO '2008 www.paco.sicpro.org, ISBN 978-5-91450-016-7.
10. Cray Will Leverage an "Adaptive Supercomputing" Strategy to Deliver the Next Major Productivity Breakthrough. SEATTLE, WA, Mar 20, 2006. <http://investors.cray.com>.
11. Cray Introduces Next-Generation Supercomputers. Seattle, Wa, Nov 06, 2007. <http://investors.cray.com/>
12. Kahle J. A. , Day M. N. , Hofstee H. P. , Johns C. R. , Maeurer T. R. , Shippy D. Introduction to the Cell multiprocessor // IBM Journal of Research and Development. VOL. 49, NO. 4/5. July/September 2005. P. 589 - 603.
13. Gaudin, Sharon. IBM set to test the fastest computer in the world. May 13, 2008. <http://www.computerworld.com>.
14. Nozawa, Tetsuo Next-Generation Supercomputer to Be Scalar/Vector Multi-System, Developed by Hitachi, NEC and Fujitsu. Jun 14, 2007. Nikkei Electronics. http://techon.nikkeibp.co.jp/english/NEWS_EN.
15. ORNL's 'Jaguar' Leaps Past Petaflop. November 11, 2008. http://www.hpcwire.com/blogs/ORNLs_Jaguar_Leaps_Past_Petaflop_34282109.html.

Системы реального времени (СРВ) с самоконтролем вычислительных средств (ВС) и схем на базовых БИС

Хетагуров Я. А., *д. т. н., профессор*

Главными факторами, определяющими системы реального времени, являются:

1. быстродействие выполнения алгоритмов и программ;
2. достоверность выдаваемой информации - отсутствие ошибок в течение заданного времени работы;
3. защищенность и безопасность;
4. длительность жизни СРВ.
5. На основе этих главных факторов оценивается влияние на затраты:
 - а. быстродействия и объемов программ ВС;
 - б. методов контроля, защищенности и обеспечения надежности;
 - в. величин потребления электроэнергии;
 - г. габаритно-массовых показателей;
 - д. ЗИПа.

Рассмотрим взаимосвязи и влияние приведенных факторов на затраты. Анализ начнем с алгоритмов решения поставленной задачи. Основное внимание должно обращаться на выбор алгоритмов, имеющих относительно низкое быстродействие, в которых предусмотрена защита и условия безопасности. Важным условием является обеспечение достоверности выдаваемой информации, которая определяется методом контроля и надежностью работы аппаратуры.

Выбор метода контроля оказывает существенное влияние на быстродействие и объем памяти. Применение микропроцессоров массового применения, выпускаемых иностранной промышленностью, предусматривает использование только программных методов контроля: что ведет к увеличению требований по быстродействию вычислительного устройства в 2,1-2,3 раза. При использовании двух вычислительных устройств или трех устройств (мажоритарная схема) требуется увеличение быстродействия только в 0,19-0,2 раза.

Однако аппаратура вычислительных средств увеличивается в 2,2 и 3,5 раза.

Использование микропроцессоров связано с применением операционной системы (ОС) и определенного языка программирования. На ОС в среднем расходуется 60-70% быстродействия микропроцессора, то есть требуется увеличение быстродействия 2,5-3,3 раза относительно необходимого для выполнения программ алгоритмов, а увеличение программ при трансляции приводит к требованию повышения быстродействия минимум на 30-60 %, а в среднем в 2-3 раза.

Обеспечение безопасности и защищенности СРВ в ряде случаев приводит к необходимости дополнительного увеличения быстродействия на 10-20%. Приведенные оценки по требованиям повышения быстродействия пропорционально увеличивают размеры памяти и объемы ее аппаратуры.

Ощутимые экономические потери СРВ наносят вирусы. Средств, предусматривающих защиту от вирусов, пока не найдено, хотя методы борьбы с выявленными вирусами известны. Они требуют дополнительных ресурсов (быстродействия и памяти), которые оцениваются в настоящее время в 5-15 % быстродействия.

В результате требования к суммарному быстродействию микропроцессора для работы в СРВ увеличиваются не менее чем в 8-17 раз по отношению к быстродействию, необходимому для выполнения рабочих программ алгоритмов. Объем памяти повышается в 6-8 раз.

Важную часть аппаратуры СРВ составляют источники электропитания. Мощность системы электропитания определяется максимальным потреблением электроэнергии в процессе работы СРВ. При использовании двоичной системы счисления характерна высокая неравномерность потребления электроэнергии в процессе работы в связи с наличием «бедных» и «богатых» единицами слов.

Существенное влияние на затраты СРВ оказывает длительность эксплуатации - длительность жизни СРВ или ее цикл жизни. При длительности

эксплуатации СРВ (обычно 10-20 лет), учитывая сроки изменения серий микропроцессоров, контроллеров и других элементов порядка 2-3 лет, необходимо предусматривать на срок эксплуатации аппаратуры в ЗИПе — 4-7 комплектов элементов.

Приведенные оценки затрат на построение СРВ основаны на применении иностранных микропроцессоров, контроллеров и других схем а также их программного обеспечения.

Использование в наших СРВ иностранных микропроцессоров и программного обеспечения практически дает возможность получения любой закрытой информации иностранными службами, то есть не обеспечивает безопасность и защищенность наших СРВ, к которым предъявляются эти требования.

Для построения СРВ, отвечающих поставленным требованиям, на основе освоенных нашей полупроводниковой промышленностью технологий, необходимо снижение требований к быстродействию аппаратуры и выполнению программ, для чего необходимо:

1. проводить исследования по созданию алгоритмов СРВ с относительно низкими требованиями к быстродействию, обеспечивающих необходимую защищенность и безопасность, что в большинстве случаев приводит к снижению требований к быстродействию в разы;
2. использовать аппаратуру с самоконтролем, которая практически исключает требование к повышению быстродействия, необходимое при программных методах контроля;
3. применять специальные команды в микропроцессоре, которые аппаратно выполняют функции операционной системы (ОС), что исключает затраты быстродействия на работу операционной системы;
4. использовать структуры построения микропроцессора и программного обеспечения, исключающих влияние вирусов, что снимает затраты на быстродействие по защите от известных вирусов;
5. выполнять требования по защищенности и безопасности аппаратными средствами, что так же снижает требования к быстродействию.

Реализация этих положений (2, 3, 4, 5) уменьшает в 8-17 раз требования к быстродействию и сокращает объемы аппаратуры в зависимости от используемых способов в среднем в 2-4 раза. Наиболее важным является положение 1, которое также сокращает требования к быстро-

действию и приводит к уменьшению объемов аппаратуры.

Выполнение этих положений приводит к построению вычислительных средств на микропроцессорах, контроллеров и других схемах с самоконтролем на основе использования существующих базовых БИС с их функциональными возможностями и быстродействием.

В результате решаются две проблемы: *первая* - обеспечивается необходимое быстродействие и уменьшаются затраты на построение вычислительных средств, разработку, производство БИС (микропроцессоров, контроллеров и других схем), а также выполняются требуемые условия работы; *вторая* - решается главное – выполнение требований по защищенности и безопасности СРВ.

Ключевой задачей является построение микропроцессора, контроллеров и других схем с самоконтролем. Для реализации самоконтроля была разработана идея использования недвоичной системы кодирования информации, известной как 1 из Р с «активным» нулем, где Р - основание системы счисления.

В результате проведенных исследований была определена рациональная величина Р равная 4. В настоящее время разработаны схемы и правила выполнения арифметических и логических операций при кодировании 1 из 4.

Приведём таблицу соответствия для двух разрядов двоичной системы и системы 1 из 4 с активным нулём.

№	Двоичная система	Система 1 из 4
1	00	0001
2	01	0010
3	10	0100
4	11	1000

Основными показателями системы кодирования 1 из 4 являются:

- прозрачная связь с двоичной системой, которая реализуется простейшими схемами. Это обеспечивает широкое использование ПО и массивов данных, созданных с использованием двоичной системы, в системе 1 из 4;
- достоверность выдаваемой информации, обеспечиваемая простотой контроля - наличием всегда одной единицы в разряде 1 из 4 и выявлением всех одиночных ошибок;

- увеличение быстродействия в 1,2-1,3 раза вследствие сокращения числа разрядов цепочки переноса в два раза, а также сокращения количества суммирований в операции умножения;
- сокращение числа ячеек памяти при одинаковых требованиях по достоверности и точности вычислений по отношению к двоичной системе с мажоритарным контролем, которое определяется величиной 1,5;
- уменьшение потребляемой мощности для микропроцессоров, построенных на КМОП технологии, при кодировании 1 из 4, по сравнению с кодированием в двоичной системе (здесь для достижения требуемой точности и достоверности информации используется мажоритарный контроль) – при максимальном потреблении снижается в 6 раз.
- повышение надежности работы микропроцессора путем резервирования (учитывая независимый контроль каждого разряда), по разрядам или группам разрядов. Это позволяет увеличивать кратность резервирования для существенного повышения надежности или сокращать объемы резервируемой аппаратуры для получения заданной надежности.

Выводы

1. Применение для СРВ микропроцессоров, контроллеров и других схем, использующих кодирование 1 из 4 схемы самоконтроля, практически исключает требование повышения быстродействия, характерное для микропро-цессоров и контроллеров на основе двоичного кодирования.
2. Применение освоенных нашей промышленностью базовых БИС ОАО «Ангстрем» обеспечивает выполнение принятых условий их эксплуатации.
3. Использование кодирования 1 из 4 по сравнению с двоичным кодированием, при обеспечении требуемой точности и достоверности вычисления (троированием), уменьшает объем аппаратуры СРВ в 2-3 раза и потребление электроэнергии в 5-6 раз, что снижает затраты на изготовление аппаратуры в 2-3 раза и в 5-6 раз ЗИПа, а главное, в 4-5 раз на эксплуатацию.
4. Важным является обеспечение требуемой защищенности и безопасности СРВ, основанное на применении отечественной аппаратуры.

Создание СРВ на основе достигнутых показателей отечественных базовых БИС (ОАО «Ангстрем») решает три принципиальные задачи:

- обеспечивает создание рациональных СРВ в относительно короткие сроки, используя базовые БИС;
- исключает возможность влияния на работу СРВ внешних факторов и обеспечивает защищенность и национальную безопасность;
- создает условия предприятиям микроэлектроники совершенствовать свою базу и повышать характеристики отечественных базовых БИС.

Новый класс сверточных кодов

Угрелидзе Н. А. , д. т. н. , профессор

Грузинский Технический Университет

Предлагается новый класс сверточных кодов, имеющих символично-алфавитную избыточность. Описаны данные коды и приведен пример кодирования симплексного сигнала таким кодом. Рассматривается вопрос сверточного кодирования, при котором повышается надежность системы независимо от того, данные коды сами по себе, исправляют ошибки или нет. Известно [1,2], что для традиционного способа кодирования (коды с символической избыточностью) в некотором интервале времени, при наличии на входе кодера k информационных символов на выходе имеется n кодовых символов, причем алфавиты входных и выходных символов совпадают и скорость кодирования $R=k/n$, $n>k$.

В нашем случае нет необходимости выполнения неравенства $n>k$ и тождественности алфавитов входных и выходных символов. При этом даже возможно, что алфавиты символов на разных входах и разных выходах кодера были разными. Такие коды были использованы ранее для улучшения спектральных характеристик сигналов [3] и повышения помехоустойчивости сигналов частотной модуляции с непрерывной фазой [4]. В обоих случаях $n=k=1$. В дальнейшем такие коды будут называться кодами с символично- алфавитной избыточностью и аксиоматически будет принято, что кодирование эффективно только для случая $0<R<1$.

Сверточное кодирование можно рассмотреть как дискретный процесс, который можно моделировать с помощью конечного A автомата [5,6]. Он имеет k входов и n выходов. На входе автомата, в заданном интервале времени, можем иметь блок символов u_1, u_2, \dots, u_k , а на выходе v_1, v_2, \dots, v_n , которые относятся к соответствующим алфавитам U_1, U_2, \dots, U_k и V_1, V_2, \dots, V_n . Кроме этих, существуют также символы, которые представляют внутреннее состояние автомата через блок $s_1, s_2, \dots, s_{\gamma+k}$ с алфавитом $S_1, S_2, \dots, S_{\gamma+k}$ (γ есть количество элементов задержки в кодере). В итоге имеем автомат $A=(u, v, s, F, \Phi)$, где F называется функцией переходов и представляет отображение $u \rightarrow s$, а Φ есть функция выходов с отображением $s \rightarrow v$. В дальнейшем через U и V будут обозначены размеры тех алфавитов, к которым принадлежат символы u и v .

Если память, которая существует в кодере, представить в виде элементов задержки, кодер будет иметь вид, представленный на рисунке.

Здесь в сумматорах и перемножителях операции выполняются по предварительно определенному закону (В отличие от классического случая, они могут быть арифметическими). При описании информационной и кодовой последовательностей через оператор задержки D будем иметь следующие кодовые генераторы:

для 1-го входа

$$G_{11}(D) = x_{110} \cdot D^0 + x_{111} \cdot D^1 + \dots + x_{11C_1} \cdot D^{C_1},$$

$$G_{12}(D) = x_{120} \cdot D^0 + x_{121} \cdot D^1 + \dots + x_{12C_2} \cdot D^{C_2},$$

... ..

$$G_{1n}(D) = x_{1n0} \cdot D^0 + x_{1n1} \cdot D^1 + \dots + x_{1nC_n} \cdot D^{C_n};$$

для 2-го входа

$$G_{22}(D) = x_{210} \cdot D^0 + x_{211} \cdot D^1 + \dots + x_{21C_1} \cdot D^{C_1},$$

$$G_{22}(D) = x_{220} \cdot D^0 + x_{221} \cdot D^1 + \dots + x_{22C_2} \cdot D^{C_2},$$

... ..

$$G_{2n}(D) = x_{2n0} \cdot D^0 + x_{2n1} \cdot D^1 + \dots + x_{2nC_n} \cdot D^{C_n};$$

⋮

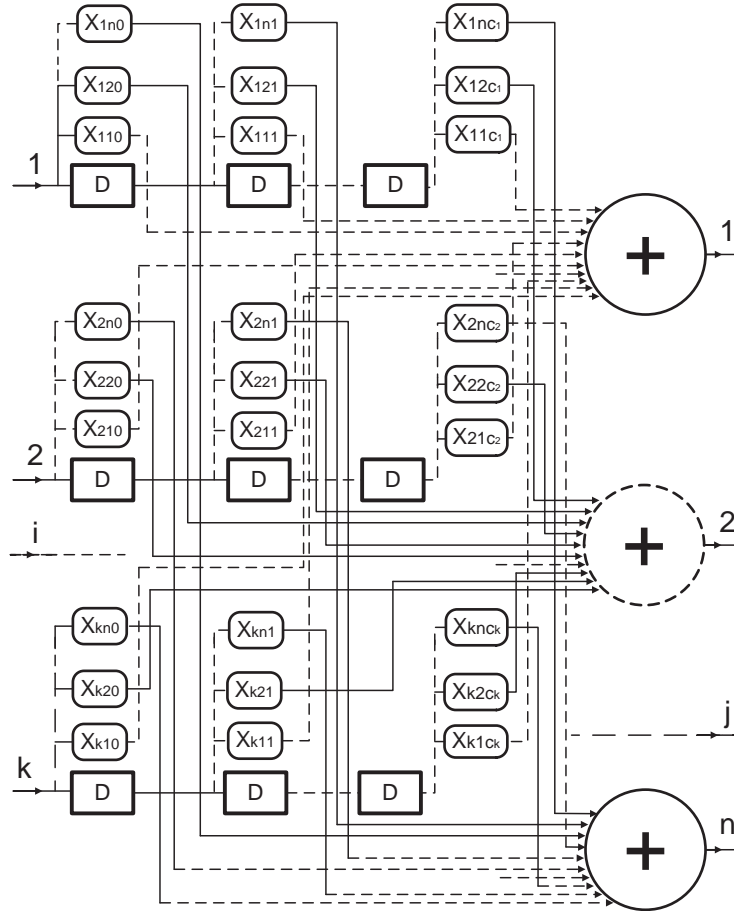
для k -го входа

$$G_{k1}(D) = x_{k10} \cdot D^0 + x_{k11} \cdot D^1 + \dots + x_{k1C_1} \cdot D^{C_1},$$

$$G_{k2}(D) = x_{k20} \cdot D^0 + x_{k21} \cdot D^1 + \dots + x_{k2C_2} \cdot D^{C_2},$$

... ..

$$G_{kn}(D) = x_{kn0} \cdot D^0 + x_{kn1} \cdot D^1 + \dots + x_{knC_n} \cdot D^{C_n}.$$



В данном рисунке и в изображениях индекс множителя X_{a,b,c_r} обозначает: a - номер входа кодера; b - номер выхода кодера; c_r - номер множителя соответствующего элемента задержки r -го входа.

Исходя из вышеизложенного, матрица кодовых генераторов будет иметь вид:

$$G(D) = \begin{pmatrix} G_{11}(D) & G_{12}(D) & \dots & G_{1n}(D) \\ G_{21}(D) & G_{22}(D) & \dots & G_{2n}(D) \\ \vdots & \vdots & \dots & \vdots \\ G_{k1}(D) & G_{k2}(D) & \dots & G_{kn}(D) \end{pmatrix}$$

С учетом того, что размеры алфавитов символов разных входов и выходов кодера могут быть разными, скорость кодирования определим так:

$$R = (\log U_1 + \log U_2 + \dots + \log U_k) / (\log V_1 + \log V_2 + \dots + \log V_n)$$

из этого окончательно получим:

$$R = \frac{\log \prod_{i=1}^k U_i}{\log \prod_{j=1}^n V_j}$$

Приведем один пример. Кодовый генератор $G(D) = [1 + D + 2D^2]$ порождает сверточный код свободным Хэмминговым расстоянием $d_f = 3$. Естественно, если таким кодом кодировать двумерный

троичный симплексный сигнал, при скорости кодирования $R=0.63$ получим квадрат свободного Евклидова расстояния сигнально-кодовой системы $d_f^2 = 4.5$.

Полагаем, что новые подклассы непрерывных кодов займут значительное место в теории и практике телекоммуникации. Исходя из этого, исследование параметров, характеристик и свойств таких кодов будет актуальным и перспективным.

Литература

1. Peterson W. W. , and Weldon E. J. Error-Correcting Codes. The MIT Press, Cambr. , Mass. , 1972.
2. Касами Т. , и др. Теория Кодирования. Пер. С яп. Под ред. А. В. Кузнецова. М. : Мир, 1973.
3. Kabal P. , Pasupathy S. Partial response signaling. IEEE Trans Commun. , v. COM-23, 1975, pp. 921-934.
4. Ugrelidze N. A. , Asanidze I. G. Coded CPFSK Modulation on the Basis of Different Input/Output Alphabet sizes Convolutional Encode. TECHINFORMI, Тбилиси, 1996, № 1118-96.
5. Гилл А. Введение в теорию конечных автоматов. М. : Наука, 1966.
6. Трахтенброт Б. А. , Барздинь Я. М. Конечные автоматы (Поведение и синтез). М. : Наука, 1970.

«Эффект МЭИ» в развитии отечественных школ вычислительной техники

Топорков В. В., *д. т. н., профессор*

Идея этой статьи была подсказана Э. М. Пройдаковым примерно год назад, когда в МЭИ проводился круглый стол «История и перспективы отечественной вычислительной техники». Это мероприятие было организовано совместно с телевидением Северного округа Москвы. Одна из тем была определена так: «Вклад выпускников МЭИ в создание научных школ и отечественной ВТ». Интересно, что первыми в СССР кафедры ВТ появились в МИФИ, МФТИ и МЭИ, но именно МЭИ дал подавляющее число известных разработчиков ВТ. Среди его выпускников ряд выдающихся ученых-основоположников отечественного компьютеростроения. Над этим «эффектом МЭИ» автору было любопытно поразмыслить. В феврале 2009 года в Математическом институте им. В. А. Стеклова РАН проходила конференция, посвященная 80-летию со дня рождения В. А. Мельникова, закончившего МЭИ. Там автору удалось пообщаться с Л. Н. Королевым, Г. Г. Рябовым, А. Н. Томилиным и узнать много нового о деятельности выпускников МЭИ в деле развития вычислительной техники. Эти впечатления автор попытался оформить в виде предлагаемой читателю статьи.

Истоки

Весомая роль Московского энергетического института в становлении и развитии отечественной вычислительной техники хорошо известна и признана. На ряде кафедр электрофизического и радиотехнического факультетов к концу 40-х – началу 50-х годов XX в. были созданы предпосылки для подготовки инженеров-вычислителей, острую необходимость в которых испытывала промышленность и оборона.

В 1951 году в МЭИ создается специальность «математические и счетно-решающие приборы и устройства». В том же году состоялся первый выпуск инженеров, поскольку соответствующая подготовка уже проводилась в рамках специальности «автоматика и телемеханика». Тогда же в 1951 году была создана кафедра «Счетно-решающие приборы и устройства» – прародительница кафедры вычислительной техники (ВТ). Ее заведующим стал Григорий Митрофанович Жданов (1898-1967), доктор технических наук, профессор. Нужно сказать, что в то время это была одна из первых кафедр в СССР, которая начала подготовку инженеров-вычислителей.

В 1955 году была организована объединенная кафедра автоматики, телемеханики и математических машин, а в 1958 году выделена кафедра ВТ, которой до 1967 года бессменно руководил ее

основатель – Г. М. Жданов. Тематикой вычислительных машин он занимался с 1937 года, в 1956 году издательство Гостехтеориздат выпустило его учебник «Математические машины и приборы непрерывного действия».

Г. М. Жданов хорошо понимал, что для будущих специалистов по вычислительной технике необходима углубленная подготовка по математическому и программному обеспечению, схемотехнике и конструированию ЭВМ. Поэтому с момента основания кафедры ВТ и в дальнейшем к преподаванию привлекались ведущие ученые и специалисты: С. А. Лебедев, М. А. Карцев, П. И. Китов, Н. Я. Матюхин, Б. И. Рамеев, И. М. Тетельбаум и др. Некоторые из них, например, Н. Я. Матюхин и М. А. Карцев, были выпускниками радиотехнического факультета МЭИ.

Огромная роль в становлении и развитии школы вычислительной техники МЭИ принадлежит академику Сергею Алексеевичу Лебедеву (1902-1974).

В 1945 г. С. А. Лебедев создал первую в стране электронную аналоговую вычислительную машину для решения систем обыкновенных дифференциальных уравнений, которые часто встречаются в задачах энергетики. Деятельность Сергея Алексеевича всегда была тесно связана с МЭИ. Довольно долго он работал на кафедре релейной защиты и автоматизации энергосистем, в 50-е

годы читал в МЭИ курс лекций «Вычислительные машины дискретного действия».

Решая задачи электротехники и энергетики с помощью аналоговых вычислительных машин, С. А. Лебедев пришел к постановке задачи создания цифровой машины. Профессор А. В. Нетушил, окончивший МЭИ за несколько лет до войны, впоследствии декан факультета автоматики и вычислительной техники, подготовил кандидатскую диссертацию на тему «Анализ триггерных элементов быстродействующих счетчиков импульсов». Он отмечает: «С самого начала этой работы в 1939 г. и до защиты С. А. Лебедев с вниманием и одобрением относился к моим исследованиям. Он согласился быть оппонентом по диссертации, защита которой состоялась в конце 1945 г. В то время еще никто не подозревал, что С. А. Лебедев вынашивает идею создания цифровой вычислительной машины».

С осени 1948 г. С. А. Лебедев начал разработку Малой электронной счетной машины (МЭСМ). Для определения набора операций МЭСМ он пригласил приехать в Киев А. А. Дородницына и К. А. Семендяева. Основы построения МЭСМ обсуждались в январе-марте 1949 г. на семинаре, в котором участвовали М. А. Лаврентьев, Б. В. Гнеденко, А. Ю. Ишлинский, А. А. Харкевич и сотрудники лаборатории С. А. Лебедева в Институте энергетики Академии наук Украины.

В конце 1951 г. МЭСМ прошла испытания и была принята в эксплуатацию Комиссией АН СССР во главе с академиком М. В. Келдышем. В состав комиссии входили академики С. Л. Соболев, М. А. Лаврентьев, профессора К. А. Семендяев, А. Г. Курош.

В 1952 г. на МЭСМ решались важнейшие научно-технические задачи из области термоядерных процессов (Я. Б. Зельдович), космических полетов и ракетной техники (М. В. Келдыш, А. А. Дородницын, А. А. Ляпунов), дальних линий электропередач (С. А. Лебедев), механики (Г. Н. Савин), статистического контроля качества (Б. В. Гнеденко).

В 1950 г. , когда был опробован макет МЭСМ, подобная машина работала лишь в Англии – ЭДСАК М. Уилкса (1949 г.), причем в ЭДСАК арифметическое устройство было последовательным.

В 1950 г. С. А. Лебедев начал разработку БЭСМ АН СССР. В марте 1950 г. он был назначен заведующим лабораторией №1 Института точной механики и вычислительной техники (ИТМ и ВТ), директором которого стал М. А. Лаврентьев.

Над БЭСМ трудились и студенты-практиканты из вузов, выполнявшие дипломные

работы: В. С. Бурцев, В. А. Мельников, А. Г. Лаут, И. Д. Визун, А. С. Федоров и Л. А. Орлов. В апреле 1951 г. Государственная комиссия под председательством М. В. Келдыша приняла эскизный проект машины БЭСМ. К концу 1951 г. основу лаборатории №1 составили выпускники МЭИ, среди которых были будущие академики В. А. Мельников и В. С. Бурцев.

В I квартале 1953 г. БЭСМ была налажена, а в апреле 1953 г. принята Государственной комиссией в эксплуатацию.

В связи с дефицитом электронных трубок, которые поставлялись тогда только для «Стрелы», первые три года БЭСМ эксплуатировалась с памятью на акустических ртутных трубках, что снижало ее быстродействие в несколько раз. В 1956 г. БЭСМ была принята Государственной комиссией вторично – с памятью на потенциалоскопах.

В 1956 г. доклад С. А. Лебедева о БЭСМ на международной конференции в Дармштадте произвел сенсацию – БЭСМ была на уровне лучших американских машин и самой быстродействующей в Европе.

В 1958 г. БЭСМ с памятью на ферритовых сердечниках емкостью 2048 слов передали в серийное производство, она выпускалась под названием БЭСМ-2.

В 1953 г. по рекомендации М. А. Лаврентьева, ставшего вице-президентом АН СССР, С. А. Лебедев был назначен директором ИТМ и ВТ.

В 1955 г. С. А. Лебедев начал разработку М-20 (цифра в названии указывала на ожидаемое быстродействие – 20 тыс. оп. /с). Такого быстродействия тогда не имела ни одна машина в мире. Постановлением Правительства СССР создание М-20 было поручено ИТМ и ВТ и СКБ-245. С. А. Лебедев стал главным конструктором, М. К. Сулим (СКБ-245) – его заместителем. Идеологию и структуру М-20 разрабатывал С. А. Лебедев, систему команд – М. Р. Шура-Бура, схемотехнику – П. П. Головистиков, М. К. Сулим руководил разработкой технической документации и изготовлением опытного образца в СКБ-245.

В 1958 г. Государственная комиссия приняла М-20 и рекомендовала ее в серийное производство.

Впервые в отечественной практике в М-20 С. А. Лебедевым с целью повышения производительности были реализованы автоматическая модификация адреса, совмещение работы арифметического устройства и выборки команд из памяти, введение буферной памяти для массивов данных, выдаваемых на печать, совмещение ввода и вывода данных со счетом, использование полностью

синхронной передачи сигналов в логических цепях.

Позднее были разработаны полупроводниковые варианты М-20, реализующие ту же архитектуру: М-220 и М-222 (главный конструктор – М. К. Сулим); БЭСМ-3М и БЭСМ-4 (главный конструктор – О. П. Васильев).

ИТМ и ВТ после завершения работ по ламповым БЭСМ-2 и М-20 начал проектирование полупроводниковой БЭСМ-6, которая обладала скоростью действия 1 млн. оп. /с. Главным конструктором БЭСМ-6 был С. А. Лебедев, заместителями – его ученики В. А. Мельников и Л. Н. Королев.

В 1967 г. Государственная комиссия под председательством М. В. Келдыша приняла БЭСМ-6 с высокой оценкой и рекомендовала ее к серийному производству.

На основе БЭСМ-6 были созданы вычислительные центры коллективного пользования для научных организаций, системы автоматизации научных исследований в ядерной физике и других областях науки, информационно-вычислительные системы обработки информации в реальном времени. Она использовалась для моделирования сложнейших физических процессов и процессов управления, в системах проектирования программного обеспечения для новых ЭВМ.

БЭСМ-6 выпускалась Московским заводом САМ в течение 17 лет. За разработку и внедрение БЭСМ-6 ее создатели (из ИТМ и ВТ – С. А. Лебедев, В. А. Мельников, Л. Н. Королев, Л. А. Зак, В. Н. Лаут, В. И. Смирнов, А. А. Соколов, А. Н. Томили, М. В. Тяпкин, от завода САМ – В. А. Иванов, В. Я. Семешкин) были удостоены Государственной премии.

ИТМ и ВТ совместно с заводом САМ на основе БЭСМ-6 разработал вычислительную систему АС-6, модульная организация и унифицированные каналы обмена которой обеспечивали возможность построения децентрализованных многомашинных вычислительных комплексов. АС-6 использовалась для обработки данных и управления в системах управления космическими поретами, а также в ряде вычислительных центров крупных научно-исследовательских организаций.

Специализированные ЭВМ, созданные под руководством С. А. Лебедева для системы противоракетной обороны (ПРО), стали основой достижения стратегического паритета СССР и США в период «холодной войны». В 1952-1955 гг. В. С. Бурцевым были разработаны специализированные ЭВМ «Диана-1» и «Диана-2» для автоматического съема данных с радиолокатора и автоматического слежения за целями. Затем для системы

ПРО, генеральным конструктором которой был Г. В. Кисунько, в 1958 г. была предложена ламповая ЭВМ М-40, а немного позднее М-50 (с плавающей точкой).

Возможность поражения баллистических ракет, обеспеченная ПРО, заставила США искать пути заключения договора с СССР об ограничении ПРО. Создатели первой системы ПРО получили Ленинскую премию. Среди них были Г. В. Кисунько, С. А. Лебедев и В. С. Бурцев.

В 50-е – 60-е годы прошлого столетия в области отечественной вычислительной техники развивалось несколько направлений. Помимо школы С. А. Лебедева, наиболее известными были научные школы В. М. Глушкова, И. С. Брука, Б. И. Рамеева. Все они серьезно влияли на становление исследований и подготовку инженеров в МЭИ.

Академик Виктор Михайлович Глушков неоднократно бывал в МЭИ, а Институт кибернетики АН УССР, основателем и директором которого он был долгое время, поддерживал тесные деловые отношения с кафедрой ВТ.

В Энергетический институт АН СССР И. С. Бруком были приглашены выпускники радиотехнического факультета МЭИ Михаил Александрович Карцев и Николай Яковлевич Матюхин, которые впоследствии преподавали на кафедре ВТ.

Башир Искандерович Рамеев, до 1938 года учившийся в МЭИ, также некоторое время работавший у И. С. Брука, стал основателем знаменитой «пензенской школы».

Таким образом, к началу 50-х годов в МЭИ был создан мощный потенциал, который во многом предопределил развитие отечественной вычислительной техники.

Выдающиеся выпускники МЭИ – основоположники отечественного компьютеростроения

Среди выпускников МЭИ много выдающихся ученых, внесших поистине огромный вклад в становление и развитие вычислительной техники.

Этот раздел хотелось бы начать с описания деятельности ярких представителей школы С. А. Лебедева. В первом же выпуске МЭИ инженеров-вычислителей в 1951 году были В. А. Мельников и В. С. Бурцев.

Владимир Андреевич Мельников (1928-1993) свой трудовой путь начал, еще будучи студентом МЭИ, в ИТМ и ВТ АН СССР под руководством академика С. А. Лебедева.

Его первой работой был макет блока центрального управления операциями машины БЭСМ. Дипломный проект В. А. Мельникова по этому блоку, наряду с проектами других студентов-практикантов В. С. Бурцева, А. Н. Зимарева, В. П. Смирягина, А. Г. Лаута, В. Н. Лаута, А. С. Федорова, составил эскизный проект БЭСМ, который был защищен в апреле 1951 г. одновременно с эскизным проектом ЭВМ «Стрела», разработанной в СКБ-245.

В. А. Мельников вырос как ученый и конструктор в научной школе академика С. А. Лебедева. До конца своей жизни Владимир Андреевич сохранил чувства глубокой любви и уважения к Учителю, вместе с соратниками и учениками он продолжил развитие школы С. А. Лебедева. В. А. Мельников вспоминал: «Мне очень повезло и с моим первым учителем, и с моим первым директором. Моим первым учителем был С. А. Лебедев, который научил нас основам создания электронных цифровых вычислительных машин, а первым директором – М. А. Лаврентьев, сумевший создать в ИТМ и ВТ творческие условия, способствовавшие разработке и организации серийного производства первых ЭВМ».

Еще будучи молодым инженером, В. А. Мельников стал одним из руководителей следующей разработки ИТМ и ВТ – ЭВМ БЭСМ-2, ее наладки и подготовки к серийному производству.

В истории отечественной вычислительной техники эпохальным событием следует считать создание ЭВМ БЭСМ-6, обладавшей средним быстродействием 1 млн. операций в секунду. Основные решения построения БЭСМ-6 (архитектура, структура машины, система элементов и схемотехника, конструкция, программное обеспечение) принадлежат главному конструктору С. А. Лебедеву, его заместителям В. А. Мельникову и Л. Н. Королеву (ныне члену-корреспонденту РАН, заведующему кафедрой в МГУ), А. А. Соколову. Многие из этих решений, реализованных на технической базе ЭВМ второго поколения, были положены в основу архитектуры ЭВМ третьего и четвертого поколений.

Архитектура БЭСМ-6 была использована в дальнейшем в ЭВМ «Эльбрус-1К2» и «Эльбрус-КБ», созданных М. В. Тяпкиным уже на интегральных схемах и обеспечивших программную совместимость с БЭСМ-6.

В 1969 г. В. А. Мельникову в составе коллектива разработчиков БЭСМ-6 была присуждена Государственная премия СССР.

В 1976 г. В. А. Мельникова избрали членом-корреспондентом АН СССР по Отделению математики.

После БЭСМ-6 В. А. Мельников, в качестве главного конструктора, вместе с С. А. Лебедевым и А. А. Соколовым, приступил к разработке вычислительной системы АС-6.

В АС-6 были воплощены многие идеи, составившие основу будущих супер-ЭВМ. АС-6 использовалась совместно с БЭСМ-6 при реализации советско-американской космической программы «Союз-Аполлон» и при последующих запусках космических кораблей в СССР.

Последние 10 лет жизни В. А. Мельников посвятил созданию векторно-конвейерных супер-ЭВМ на отечественной элементной базе, наиболее полно отвечающих требованиям решения сложных задач вычислительной математики. С этой целью он создал коллектив разработчиков супер-ЭВМ в одном из научно-исследовательских институтов Минэлектронпрома СССР. В 1983 г., когда в АН СССР было открыто Отделение информатики, вычислительной техники и автоматизации, В. А. Мельников организовал Институт проблем кибернетики АН СССР и стал его директором. Вычислительная система «Электроника-ССБИС», которая разрабатывалась В. А. Мельниковым, в архитектурном отношении напоминала известную систему фирмы Cray Research (США), хотя конкретные решения по архитектуре, конструкции, схемотехнике и особенностям системного программного обеспечения для векторно-конвейерной супер-ЭВМ были оригинальными.

В 1986 г. В. А. Мельников был избран действительным членом АН СССР по Отделению математики.

Начавшаяся в СССР в конце 80-х годов перестройка, а затем длительный экономический кризис не позволили В. А. Мельникову довести до конца работу по «Электронике-ССБИС».

Большая и плодотворная работа В. А. Мельникова была отмечена высокими наградами – орденом Ленина, двумя орденами Трудового Красного Знамени и медалями. В. А. Мельников – дважды лауреат Государственных премий (1969 и 1980 гг.), лауреат премии им. С. А. Лебедева Президиума АН Украины.

Академик РАН Всеволод Сергеевич Бурцев является крупнейшим специалистом в области создания высокопроизводительных вычислительных машин и комплексов. Он родился 11 февраля 1927 г. Еще до окончания Московского энергетического института Всеволод Сергеевич начал на-

учную и инженерную деятельность в ИТМ и ВТ под руководством академика С. А. Лебедева. Темой его дипломной работы была система управления БЭСМ АН СССР. Уже на дипломном проектировании он стал одним из ведущих разработчиков.

В 1953-1956 гг. В. С. Бурцев, будучи ответственным исполнителем, предложил принцип селекции и оцифровки радиолокационного сигнала. На его основе был осуществлен съем данных о цели с радиолокационной станции и ввод их в вычислительную машину, разработаны специализированные вычислительные машины «Диана-1» и «Диана-2», успешно проведен эксперимент одновременного сопровождения нескольких целей вычислительной машиной. На базе этих работ В. С. Бурцев написал кандидатскую диссертацию. На защите диссертации члены совета единогласно проголосовали за присуждение В. С. Бурцеву ученой степени доктора технических наук, так как полученные им результаты в корне изменили структуру управляющих противоракетных и противосамолетных комплексов.

В 1956-1961 гг. под непосредственным руководством Всеволода Сергеевича были разработаны принципы построения вычислительных средств ПРО страны и создан двухмашинный высокопроизводительный вычислительный комплекс на базе спроектированных в ИТМ и ВТ машин М-40 и М-50.

Для М-40, самой быстродействующей в то время в стране серийной машины, В. С. Бурцевым впервые были предложены принципы распараллеливания вычислительного процесса на уровне аппаратных средств.

М-50 была выполнена как модификация М-40 для выполнения арифметических операций с плавающей точкой. Она была введена в эксплуатацию в 1959 г.

За создание вычислительных комплексов ПРО на базе М-40 и М-50 С. А. Лебедев и В. С. Бурцев были удостоены Ленинской премии.

В 1961-1968 гг. под непосредственным руководством В. С. Бурцева были разработаны высокопроизводительные полупроводниковые машины, предназначенные для построения сложных боевых комплексов ПРО: 5Э926 и ее модификация для вычислений с плавающей точкой 5Э51. 5Э926 обладала повышенной структурной надежностью и достоверностью выдаваемой информации, основанными на полном аппаратном контроле вычислительного процесса. Многомашинный вычислительный комплекс ПРО из восьми машин 5Э926 с автоматическим резервированием

был испытан в реальной работе в 1967 г. В дальнейшем серийные машины 5Э926 стали основой системы ПРО страны, обеспечив паритет с США в «холодной войне» и сыграв важнейшую политическую роль в заключении в 1972 г. договора по ограничению ПРО.

В 1969-1972 гг. В. С. Бурцев, являясь главным конструктором, создал первую бортовую вычислительную машину третьего поколения для боевого возимого зенитно-ракетного комплекса С-300.

В декабре 1976 г. В. С. Бурцев был избран членом-корреспондентом АН СССР по Отделению механики и процессов управления.

В 1973-1985 гг. Всеволод Сергеевич руководил разработкой многопроцессорных вычислительных комплексов (МВК) «Эльбрус-1», «Эльбрус-2», являясь главным конструктором. При проектировании МВК «Эльбрус-2» по его инициативе и при непосредственном участии были созданы новые быстродействующие интегральные схемы, высокочастотные групповые разъемы, многокристальные и большие интегральные схемы, микрокабели, прецизионные многослойные печатные платы. Это было большим вкладом в развитие технологии в нашей стране. В 1980 г. были закончены работы по созданию МВК «Эльбрус-1» общей производительностью 15 млн. оп. /с. В 1985 г. успешно завершены государственные испытания десятипроцессорного МВК «Эльбрус-2» производительностью 125 млн. оп. /с. Оба комплекса были освоены в серийном производстве.

В 1986-1993 гг. под руководством В. С. Бурцева разработана архитектура супер-ЭВМ, основанная на принципе управления потоком данных, обеспечивающая распараллеливание вычислительного процесса на аппаратном уровне.

В июне 1992 г. В. С. Бурцев был избран действительным членом РАН по Отделению информатики, вычислительной техники и автоматизации.

В. С. Бурцев являлся научным руководителем фундаментальных исследований по разработке нетрадиционных архитектурных решений высокопроизводительных вычислительных машин с использованием новых физических принципов, а также их системного программного обеспечения.

Всеволод Сергеевич Бурцев удостоен Ленинской и Государственной премий, награжден орденами Ленина, Октябрьской революции, Трудового Красного Знамени и медалями. За цикл работ «Теория и практика создания высокопроизводительных многопроцессорных вычислительных

машин» ему присуждена премия им. С. А. Лебедева АН СССР.

Член-корреспондент РАН Юрий Иванович Митропольский является выпускником кафедры вычислительной техники МЭИ 1958 года. С 1963 г. работал в ИТМ и ВТ, принимал участие в разработке и внедрении БЭСМ-6, занимался проектированием системы обмена с внешними накопителями. Следующей разработкой была система обработки данных АС-6. Ю. И. Митропольский отвечал за создание периферийного процессора и системы автоматизации проектирования АПАС.

В 1978 г. вместе с академиком В. А. Мельниковым перешел в НИИ «Дельта», был назначен заместителем, а в 1981 г. – первым заместителем главного конструктора системы «Электроника ССБИС». В 1989 г. были проведены испытания головного образца системы.

В 1990 г. Ю. И. Митропольский избран членом-корреспондентом АН СССР. Участвовал в создании Института проблем кибернетики, в 1991 г. был назначен заведующим отделением этого института.

В 1994 г. был переведен на должность заместителя директора Института высокопроизводительных вычислительных систем РАН. С 1999 г. Ю. И. Митропольский заведует отделом в Институте системного анализа РАН. Под его руководством продолжают исследования по мультикомпьютерным вычислительным суперсистемам. С 1994 г. Ю. И. Митропольский возглавляет базовую кафедру Московского физико-технического института.

Михаил Александрович Карцев и Николай Яковлевич Матюхин принадлежат к школе И. С. Брука.

Михаил Александрович Карцев (1923-1983) – выпускник радиотехнического факультета МЭИ. Будучи студентом 5-го курса, работал по совместительству в Лаборатории электросистем Энергетического института АН СССР. Здесь М. А. Карцев принял участие в разработке электронной вычислительной машины М-1, проводившейся под руководством члена-корреспондента АН СССР И. С. Брука группой выпускников МЭИ во главе с Н. Я. Матюхиным.

В 1952 г. М. А. Карцев был направлен на постоянную работу в лабораторию И. С. Брука, где возглавил создание машины М-2. Небольшой коллектив справился с заданием всего за полтора года и в 1953 г. М-2 была введена в эксплуатацию.

По основным характеристикам М-2 находилась в одном ряду с созданными в 1953 г. ЭВМ «Стрела» и БЭСМ. В течение определённого времени эти три ЭВМ удовлетворяли потребности многих научных и конструкторских организаций в СССР.

Зимой 1954 - 1955 гг. М-2 была существенно модернизирована М. А. Карцевым. При модернизации М-2 в системе команд, вероятно, впервые была реализована идея укороченных адресов, послужившая предшественницей принципа формирования исполнительных адресов в ЭВМ второго и третьего поколений.

В 1958 г. была издана монография М. А. Карцева «Арифметические устройства электронных цифровых машин», позднее переизданная за рубежом.

В 1957 г. И. С. Брук начал разработку электронной управляющей машины М-4, предназначенной для управления новым экспериментальным радиолокационным комплексом, который создавался Радиотехническим институтом АН СССР под руководством академика А. Л. Минца. Руководителем разработки М-4 был назначен М. А. Карцев. К этому времени электронной промышленностью были освоены и выпускались первые отечественные транзисторы. Поэтому было решено проектировать М-4 на полупроводниковой элементной базе, и она стала одной из первых отечественных машин второго поколения.

В М-4 М. А. Карцев одним из первых реализовал разделение оперативной памяти машины на память данных и память программ и констант, размещаемых в постоянном запоминающем устройстве, для того, чтобы повысить устойчивость к отказам и сбоям систем, построенных на базе этой машины.

В июле 1962 г. были завершены совместные испытания М-4 с устройством первичной обработки информации и сопряжения с объектом на экспериментальном радиолокационном комплексе. В ноябре 1962 г. вышло постановление о запуске М-4 в серийное производство.

Хотя постановление о запуске М-4 в серийное производство было принято, М. А. Карцев, поддержанный коллективом разработчиков, настоял на проведении её существенной модернизации, учитывая прогресс в технической базе. Это позволило почти на порядок увеличить быстродействие машины. В 1964 г. появилась машина М-4М, быстродействие которой составляло 220 тыс. оп. /с. Она выпускалась до 1985 г. , всего было поставлено на объекты более 100 комплектов.

На базе М-4М и М-4-2М, включавшей в себя внешнее вычислительное устройство и средства сопряжения с каналами связи, были построены многомашинные вычислительные комплексы, объединённые в мощную вычислительную систему, работавшую в реальном масштабе времени.

В 1967 г. за эти работы М. А. Карцеву была присуждена Государственная премия СССР.

В том же году М. А. Карцев выдвинул дерзкий проект вычислительного комплекса М-9 с производительностью порядка миллиарда оп. /с. Принципиально новым был взгляд на архитектуру комплекса, когда обеспечивалась высокая степень параллелизма и высокая производительность при работе с разнородными данными для широкого класса вычислительных задач.

Эти оригинальные идеи М. А. Карцева, опередившие своё время, не были реализованы в полном объеме прежде всего из-за отсутствия тогда необходимой технической базы.

Разработка векторной вычислительной машины М-10 на микроэлектронной элементной базе была начата М. А. Карцевым в 1969 г. В 1973 г. первый промышленный образец М-10 успешно выдержал испытания. С этого времени началось серийное производство М-10, в течение 15 лет было выпущено несколько десятков этих комплексов. М-10 представляла собой многопроцессорную вычислительную систему синхронного типа и относилась к ЭВМ третьего поколения. Основное применение М-10 нашла в системе раннего предупреждения о ракетном нападении, а также в системе общего наблюдения за околоземным космическим пространством, состоявшей из сети радиолокационных станций, сопряжённых с вычислительными комплексами М-10. Разработка М-10 была проведена НИИ вычислительных комплексов (НИИВК), созданном М. А. Карцевым в 1967 г.

Впервые в мире на модели, рассчитанной с помощью М-10, физиками были получены данные по явлению коллапса в плазме, чего не удалось сделать учёным США на СДС 7600.

В 1978 г. М. А. Карцев начал разработку новой многопроцессорной векторной вычислительной машины М-13 на больших интегральных схемах. М-13 была первой в СССР векторно-конвейерной ЭВМ.

М. А. Карцев – автор фундаментальных работ по вычислительной технике, в том числе пяти монографий по арифметике и архитектуре электронных цифровых машин. В книге «Вычислительные системы и синхронная арифметика» (1978 г.) им практически впервые было

поставлено на научную основу проектирование общей структуры ЭВМ для выполнения параллельных вычислений.

М. А. Карцев был награждён орденами Ленина, Трудового Красного Знамени, "Знак почёта", орденом Красной Звезды, медалью «За отвагу» и другими медалями.

Член-корреспондент РАН Николай Яковлевич Матюхин (1927-1984) – выдающийся учёный в области вычислительной техники и конструктор специализированных ЭВМ для систем противовоздушной обороны (ПВО). Он был одним из наиболее ярких представителей инженерной школы И. С. Брука, ведущим разработчиком первых ЭВМ М-1 и М-3.

В начале 1950 г. Н. Я. Матюхин был приглашен на работу в лабораторию И. С. Брука. Там он возглавил создание первой автоматической цифровой вычислительной машины М-1, разработка которой была завершена в 1951 г. практически одновременно с созданием С. А. Лебедевым машины МЭСМ в Киеве.

Уже в этой работе Н. Я. Матюхиным были предложены принципиальные решения по выбору архитектуры и схемотехники ЭВМ первого поколения, развитые им в машине М-3 и сыгравшие важную роль в становлении отечественной вычислительной техники.

В 1956 г. группа Н. Я. Матюхина совместно с группой Б. М. Кагана из Института электропромышленности академика А. Г. Иосифьяна (ныне ВНИИЭМ) завершила разработку малой ЭВМ М-3, предназначенной для научных и инженерных расчетов в научно-исследовательских и проектно-конструкторских организациях и послужившей прототипом широко известных серий ЭВМ «Минск» и «Раздан».

В 1957 г. Н. Я. Матюхин предложил принцип микропрограммного управления ЭВМ независимо от работ М. Уилкса в Манчестерском университете (Англия). Первый макет такого управления под руководством Н. Я. Матюхина был выполнен летом 1957 г. Ю. Н. Глуховым и Е. Н. Филиновым в Лаборатории управляющих машин и систем (ЛУМС) АН СССР, преобразованной позже в Институт электронных управляющих машин (ИНЭУМ).

В 1957 г. группа сотрудников ЛУМС АН СССР (Н. Я. Матюхин, А. Б. Залкинд, О. В. Росницкий, А. И. Щуров) была приглашена в институт, возглавлявшийся тогда генеральным конструктором Г. Л. Шориным, для создания цифровых вычислительных и управляющих комплексов

для системы ПВО страны, по назначению аналогичной американской системе «Сейдж».

Первой разработкой Н. Я. Матюхина в этом направлении была специализированная ЭВМ «Тетива», выпущенная Минским заводом в 1962-1964 гг.

ЭВМ «Тетива» была первой отечественной ЭВМ второго поколения, в которой устройство управления использовало микропрограммы, хранящиеся в постоянной памяти. Позже микропрограммное управление было применено во многих отечественных ЭВМ («Проминь», 1963 г. ; «Наири», 1964 г. ; «Мир», 1965 г. ; моделях ЕС ЭВМ и СМ ЭВМ).

В дальнейшем Н. Я. Матюхин был главным конструктором ряда стационарных и возимых ЭВМ, а также вычислительных комплексов для центров коммутации сообщений глобальной сети системы ПВО, разработанных НИИ автоматической аппаратуры (НИИАА) и серийно выпускавшихся отечественной промышленностью с 1965 по 1992 гг.

Будучи главным конструктором ряда крупных проектов по созданию ЭВМ, Н. Я. Матюхин одним из первых почувствовал острую необходимость в автоматизации проектирования средств вычислительной техники. Начиная с 1964 г. , он выполнил ряд основополагающих исследований и разработок в этом направлении. Под руководством и при непосредственном участии Н. Я. Матюхина в 1968 г. была написана первая отечественная монография «Применение ЦВМ для проектирования цифровых устройств». В ней были выдвинуты и обоснованы принципы построения САПР средств вычислительной техники, лежащие ныне в основе многих конкретных систем. Н. Я. Матюхин разработал язык моделирования цифровых устройств МОДИС и первую систему моделирования. Комплексный подход к проектированию, объединяющий логическое моделирование с автоматизированным конструированием устройств и выпуском документации, также был предложен Н. Я. Матюхиным.

Н. Я. Матюхин руководил межведомственной координацией оборонных министерств СССР в области САПР вычислительной техники, всесоюзными научными конференциями и семинарами по этой проблеме.

Как крупный специалист в области вычислительной техники, один из тех, кто заложил основы этой отрасли в СССР, в 1979 г. Н. Я. Матюхин был избран членом-корреспондентом АН СССР по Отделению механики и процессов управления. В 1976 г. за работы в области систем управления

Н. Я. Матюхин был удостоен Государственной премии СССР.

Научно-исследовательскую работу Н. Я. Матюхин успешно совмещал с педагогической. Он преподавал в МЭИ, был профессором базовой кафедры Московского института радиотехники, электроники и автоматики при НИИАА.

В 1980 г. за высокие заслуги он был награжден орденом Трудового Красного Знамени.

Среди других известных выпускников МЭИ такие первоклассные специалисты как М. А. Ларионов, Г. П. Лопато, В. В. Пржиялковский, В. К. Левин и В. Н. Наумов.

Александр Максимович Ларионов (1928-1995) окончил Московский энергетический институт по специальности «электромеханические приборы» в феврале 1951 г. , был направлен в СКБ-245 (Москва), где занимался наладкой и отработкой устройства центрального управления ЭВМ «Стрела», быстро став ведущим специалистом.

В октябре 1969 г. А. М. Ларионов назначается заместителем директора НИЦЭВТ по научной работе. Одновременно он становится заместителем Генерального конструктора ЕС ЭВМ и заместителем главного конструктора бортовых ЦВМ серии «Аргон».

С марта 1971 года А. М. Ларионов – директор НИЦЭВТ, Генеральный конструктор ЕС ЭВМ, главный конструктор БЦВМ серии «Аргон».

В 1971 г. он был награжден орденом Трудового Красного Знамени, в 1977 г. – орденом Ленина.

Георгий Павлович Лопато (1924-2003) закончил электрофизический факультет МЭИ.

Георгий Павлович Лопато был главным конструктором первой ЭВМ, разработанной в СКБ завода им. Г. К. Орджоникидзе, – ламповой ЭВМ «Минск-1» (1960 г.). Наиболее значительной работой, проведенной в НИИЭВМ под его непосредственным техническим руководством, было создание по заказу Министерства обороны ряда возимых ЭВМ, совместимых с машинами ЕС ЭВМ (1983–1989гг.)

В течение длительного времени Георгий Павлович был заведующим кафедрой Минского радиотехнического института.

В 1979 г. Г. П. Лопато избран членом-корреспондентом АН СССР, в 1995 г. – членом-корреспондентом НАН Белоруссии. В 1980 г. он получил звание профессора. В апреле 1999 г. Ин-

ternational Biographical Center, Cambridge (Англия) включил Г. П. Лопато в состав 2000 выдающихся деятелей науки 20-го столетия.

В 1970 г. за вклад в разработку семейства ЭВМ «Минск» второго поколения Г. П. Лопато присуждена Государственная премия. Награжден орденами Ленина (1983 г.), Октябрьской Революции (1972 г.), Трудового Красного Знамени (1976 г.), Знак Почета (1966 г.), девятью медалями.

Виктор Владимирович Пржиялковский родился второго марта 1930 г. В 1953 г. окончил факультет электровакуумной техники и специального приборостроения Московского энергетического института по специальности «автоматические и измерительные приборы и устройства».

С 1971 по 1977 гг. Виктор Владимирович – заместитель директора по научной работе НИЦЭВТ, заместитель Генерального конструктора ЕС ЭВМ, заместитель главного конструктора БЦВМ комплекса «Аргон».

В 1977 г. он назначен директором НИЦЭВТ, а затем до 1988 г. был генеральным директором НПО «Персей», Генеральным конструктором ЕС ЭВМ, главным конструктором БЦВМ комплекса «Аргон».

В 1988–1990 гг. В. В. Пржиялковский – Генеральный конструктор НПО «Персей», Генеральный конструктор ЕС ЭВМ, главный конструктор БЦВМ комплекса «Аргон».

В 1970 г. он стал лауреатом Государственной премии СССР, в 1983 г. Героем Социалистического труда. Виктор Владимирович награжден орденами Трудового Красного Знамени (1971 г.), Октябрьской Революции (1977 г.), орденом Ленина, четырьмя медалями.

Владимир Константинович Левин родился 5 марта 1929 г. Поступив в 1937 г. в среднюю школу, он досрочно окончил ее в 1944 г., сдав экзамены экстерном. В том же году он поступил в Московский энергетический институт, который окончил в 1950 г. по специальности «автоматика и телемеханика».

С 1951 по 1968 г. Владимир Константинович работал в Конструкторском бюро промышленной автоматики (КБПА) Министерства радиопромышленности, с 1960 г. – в должности главного инженера предприятия.

В 1951–1958 гг. он участвовал в создании высокопроизводительных специализированных вы-

числительных машин на лампово-диодной схемотехнике с тактовой частотой 1,5 МГц, рекордной для того времени. В 1958–1965 гг. был заместителем главного конструктора полупроводниковой высокопроизводительной вычислительной машины общего назначения «Весна» и ее расширенной модификации – «Сигма». Сокращенный вариант «Весны» – «Снег» и специализированные системы на той же элементно-конструктивной базе В. К. Левин разрабатывал в ранге главного конструктора.

В 1966–1967 гг. Владимир Константинович руководил аванпроектом системы «Ряд» (с 1968 г. именовавшейся Единой системой ЭВМ). В связи с последовавшим развертыванием работ по ЕС ЭВМ, он был переведен в 1968 г. с частью сотрудников КБПА во вновь образованный НИЦЭВТ, где до 1976 г. занимал должность заместителя директора по научной работе. В качестве первого заместителя Генерального конструктора ЕС ЭВМ он курировал вопросы разработки высокопроизводительных машин, системного программного обеспечения, работы по сотрудничеству со странами – членами СЭВ, внедрения ЕС ЭВМ в автоматизированные системы управления и информационного обслуживания с использованием средств телеобработки данных.

В 1976–1995 гг. В. К. Левин – директор КБПА, которое в 1978 г. преобразовано в НИИ «Квант». С 1996 г. по настоящее время он научный руководитель НИИ «Квант». Главный конструктор трех очередей проблемно-ориентированных систем с массовым структурным параллелизмом с наивысшими в отечественной практике показателями производительности. В процессе этих работ совместно с ИПМ им. М. В. Келдыша РАН на основе микропроцессоров Intel 860 и транспьютеров в 1993–1995 гг. создана масштабируемая многопроцессорная вычислительная система МВС-100, используемая в ряде организаций страны. В 1996–1999 гг. под руководством В. К. Левина на основе микропроцессоров Alpha DEC создана вычислительная система нового поколения МВС-1000.

В 1987 г. В. К. Левин избран членом-корреспондентом АН СССР, с 2003 г. действительный член РАН

Владимир Константинович лауреат Ленинской (1957 г.) и Государственной (1983 г.) премий, награжден орденом Ленина (1955 г.), двумя орденами Трудового Красного Знамени (1958 г. и 1971 г.), несколькими медалями.

Борис Николаевич Наумов (1927-1988) закончил МЭИ по специальности «автоматика и телемеханика».

В 1950-1967 гг. Б. Н. Наумов трудился в Институте автоматики и телемеханики. Здесь, в коллективе ведущих ученых в области автоматического управления, Б. Н. Наумов сформулировал свою позицию и выполнил исследования, направленные на создание новых эффективных методов теории нелинейных систем автоматического управления и ее приложений к решению крупных задач народного хозяйства и оборонной техники.

К 1967 г. у него уже были работы, нашедшие международное признание, блестящие защиты кандидатской и докторской диссертаций, свое место в теории автоматического управления и ее приложениях на практике. Как раз в это время оказался в кризисном положении коллектив ИНЭУМ, созданный в 1958 г. в составе АН СССР и переданный в ведение Минприбора СССР в 1965 г. Б. Н. Наумов без колебаний принял предложение руководителей Минприбора и возглавил в 1967 г. ИНЭУМ.

Наиболее ярко масштабность и организационный талант Б. Н. Наумова проявились в период 1974-1984 гг., когда он руководил разработкой международной системы малых ЭВМ (СМ ЭВМ) в качестве Генерального конструктора.

За создание СМ ЭВМ Б. Н. Наумов был удостоен Государственной премии СССР в области науки и техники. СМ ЭВМ в совокупности с Единой системой ЭВМ в 70-80-х годах послужили основой автоматизации управления и обработки информации во всех отраслях народного хозяйства, автоматизации научных исследований.

В 1976 г. Б. Н. Наумов был избран членом-корреспондентом АН СССР, в 1984 г. – действительным членом АН СССР. Б. Н. Наумов был одним из инициаторов организации в составе АН СССР Отделения информатики, вычислительной техники и автоматизации, созданного в 1983 г. усилиями вице-президента АН СССР академика Е. П. Велихова. До конца жизни Б. Н. Наумов возглавлял Институт проблем информатики АН СССР, организованный по его инициативе.

Исследования в области информатики и вычислительной техники в МЭИ

Отсчет хронологии вычислительной техники в МЭИ принято считать 1951 год, когда была открыта специальность «математические и счетно-решающие приборы и устройства» и состоялся первый выпуск инженеров-вычислителей (группа ВП-1-45). У истоков создания вычисли-

тельного направления в МЭИ стояли энергичные и талантливые люди. Работая в Киеве в Институте энергетики АН УССР, С. А. Лебедев еженедельно приезжал в Москву и читал лекции в МЭИ. Такой режим длился полтора года, затем дисциплину «Вычислительные машины дискретного действия» стал вести Анатолий Георгиевич Шигин (1922-1997). Он сделал очень много для создания лабораторной базы, а в 1952 году защитил одну из первых в СССР кандидатских диссертаций, посвященных созданию устройств ЭВМ. Исследования в области вычислительной техники в МЭИ с начала 50-х годов прошлого столетия начинают динамично развиваться. В 1953 году организуется подготовка аспирантов по соответствующим специальностям. В 1957 году приобретает массово выпускавшаяся в стране ЭВМ «Урал-1». Быстрый рост объема заявок привел к необходимости приобретения еще нескольких машин и к созданию в 1958 году нового подразделения – вычислительного центра МЭИ. С 1965 года кафедра ВТ, наряду с инженерами по ЭВМ, начинает выпуск специалистов по прикладной математике.

С 1967 по 1982 год кафедру ВТ возглавлял Юрий Матвеевич Шамаев (1922-1998). С его приходом на кафедре начинаются исследования, связанные с проектированием устройств памяти. В 60-е – 70-е годы одной из самых больших по составу на кафедре ВТ была научная группа А. Г. Шигина. По инициативе Ю. М. Шамаева в 1971 году кафедра ВТ начинает подготовку инженеров по конструированию и производству электронно-вычислительной аппаратуры. В 1976 году на базе отдела математического обеспечения создается кафедра прикладной математики (ПМ). В начале 80-х годов часть сотрудников кафедры ВТ была переведена на реорганизованную кафедру системотехники (впоследствии ставшую называться кафедрой вычислительных машин, систем и сетей – ВМСС). С 1982 по 1996 год кафедрой ВТ руководил Гурам Семенович Чхартишвили, а начиная с 1996 года заведующим кафедрой является автор этих строк.

Исследования на факультете автоматики и вычислительной техники (АВТФ), благодаря тому заделу, который создали основатели вычислительного направления в МЭИ, всегда проводились на высоком уровне и завоевали признание не только в стране, но и за рубежом. К преподаванию дисциплин по вычислительной технике в МЭИ всегда привлекались высококлассные специалисты, активно работавшие в науке. Здесь нужно вспомнить Илью Марковича Тетельбаума

(1910-1992), который работал в МЭИ с 1950 по 1980 год, а на кафедре ВТ – с момента ее основания. Фактически он был одним из основоположников аналогового моделирования в СССР. С 1956 по 1970 год на кафедре преподавал Анатолий Иванович Китов, один из ведущих ученых в области создания информационно-поисковых систем и алгоритмических языков. Дмитрий Александрович Поспелов, известный ученый в области создания ЭВМ новой архитектуры и проблем искусственного интеллекта, работал в МЭИ, в том числе и на кафедре ВТ, с 1956 по 1968 год. В начале 70-х годов преподавал на кафедре ВТ Эдуард Владимирович Евреинов, видный специалист по вопросам проектирования многопроцессорных систем. Исследования на АВТФ всегда велись широким фронтом, по практически всем известным актуальным проблемам в области информатики и вычислительной техники.

Вычислительное направление в МЭИ создано и существует уже более 50 лет. За это время подготовлено огромное число квалифицированных специалистов. Научно-педагогическая школа вычислительной техники МЭИ хорошо известна не только в нашей стране, но и за рубежом. Хочется надеяться на то, что основы и традиции, заложенные ее создателями, а также самоотверженный труд коллег будут способствовать возрождению и укреплению позиций компьютерной индустрии в России.

From Science to Business and back

by Siegfried Dickhoven

Rise and Fall of a Scientific Entrepreneurial Adventure
- The Case of vsatnet. com AG -



The Lecturer: Siegfried Dickhoven (65), Cologne, Germany

Professional education in el. engineering and management science at RWTH Aachen, Univ. of Cologne

Professional experience as Senior Researcher at DLR, GMD, Fraunhofer in the fields: modelling, applied IT, networks, satellite communication, research management and
Founder (CoB) of vsatnet. com AG
Current affiliations: Guest- Lecturer at GTU

Creating vsatnet. com AG from science

History:

1990: A new paradigm for European research centers for more competition and for better links with industry

1990-95: First multimedia communication experiments via satellite (a. o. : with RAS Researchers)

1995-99: Creating demonstrators and experimental markets (a. o. : Internet backbone to RAS) incentives and support for new technology markets

Spring 1999: Business plan development with help from GMD Technopark

Summer 1999: Finding a person to run the business

Fall 1999: Last market test on IBC Amsterdam

12/1999: Creation of vsatnet. com AG as legal business entity and signing 4 years support contract with GMD

1/2000: First customer service contract

11/1999 - 3/2000: writing proposal for startup support by ESA

7/2000 - 9/2001: ESA Piloting Project „satXpress“

3/2002 - 11/2002: ESA follow up projects HRG, SI-NA

2002: AA contract

2002: GMD/Fraunhofer merger

2004: End of contract with GMD/ Fraunhofer

2004-2007: Liquidation of vsatnet. com

The Founders:

Siegfried Dickhoven, director GMD. NET

Helmut Jungblut, head GMD. NET. SatCom

Franz Köpke, head GMD. NET. Networks

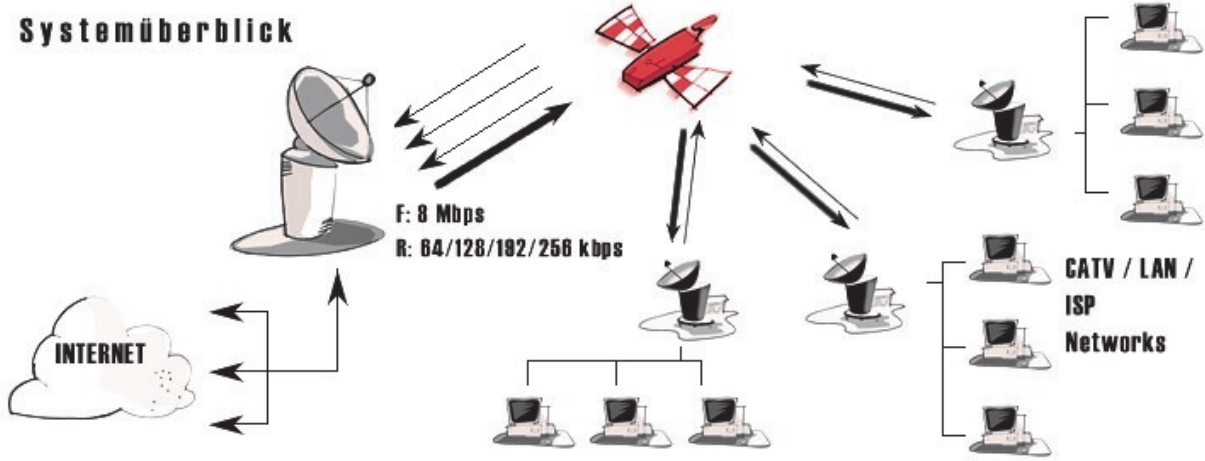
Ralf Stupp, entrepreneur from outside

Legal Form:

Aktiengesellschaft (joint stock company)

GMD Spinoff vsatnet. com AG: satXpress IP-Services

Systemüberblick



GMD Spinoff vsatnet. com AG



AG IP Networking and Beyond

satXpress

IP business services via satellite

- ① **satXpress:**
Internet access service for CATVs, LANs, regional ISPs
- ② **intra-satXpress:**
Intranet/VPN services with adaptable security
- ③ **satXpress-multicast/-broadcast:**
IP caching and IP content distribution services
- ④ **satXpress-IP-SNG:**
IP content gathering and aggregation services

powered by **SHIRON** Satellite Communications **InterSKY™**

www.vsatnet.com

Internet Carrier Services via Satellite

satXpress pricelist

Date: 1.11.2002

**Price information (EU)****satXpress**

compact satXpress backchannel CIR 64Kbps Forward up to 8Mbps	EURO
Monthly fee:	950,00
Monthly traffic volume	Per Gbyte:
up to 5 Gbyte/Month	99,00
up to 20 Gbyte/Month	81,00
more than 20 Gbyte/Month	72,00

baseline satXpress backchannel CIR 128Kbps Forward up to 8Mbps	EURO
Monthly fee:	1.450,00
Monthly traffic volume	Per Gbyte:
up to 5 Gbyte/Month	99,00
up to 20 Gbyte/Month	81,00
up to 50 Gbyte/Month	72,00
more than 50 Gbyte/Month	54,00

double satXpress backchannel CIR 256Kbps Forward up to 8Mbps	EURO
Monthly fee:	1.950,00
Monthly traffic volume	Per Gbyte:
up to 20 Gbyte/Month	81,00
up to 50 Gbyte/Month	72,00
up to 100 Gbyte/Month	54,00
more than 100 Gbyte/Month	45,00

Service type: oneway satXpress (w/out satellite backchannel)	EURO
Monthly fee:	450,00
Monthly traffic volume	Per Gbyte:
up to 5 Gbyte/Month	99,00
up to 20 Gbyte/Month	81,00
up to 50 Gbyte/Month	72,00
up to 100 Gbyte/Month	54,00
more than 100 Gbyte/Month	45,00

satXpress-multicast

	EU
	Euro
registration fee	220
space on multicast- server	
monthly fee per Gbyte	60
delivery via Internet	
fee per Gbyte	40
distribution fee per Gbyte	
bandwidth: 200 Kbps CIR	
a) immediate delivery	450
b) 24 hours delivery	225
c) weekend delivery	135
bandwidth: 2Mbps CIR	
a) immediate delivery	990
b) 24 hours delivery	495
c) weekend delivery	270

CIR (Comitted Information Rate) Services

EURO/ month	Forwardchannel					upon request
	Kbps	128	256	512	1024	
	64	1.540,00	2.380,00	3.920,00	6.930,00	
	128	2.030,00	2.940,00	4.480,00	7.420,00	
	256	3.150,00	3.850,00	5.530,00	8.400,00	

Creating vsatnet. com

Revenues	(profits)	(k€)
2000:	248	(+17)
2001:	606	(+283)
2002:	738	(+173)
2003:	898	(+171)
2004 (1st Qu.):	405	(+39)
2007:	62	(+19)

Contracted services

by type of service:

	CATV/ISP	Intra	Multi	SNG
2000	18	1	-	3
2001	16	4	1	2
2002	22	4	1	-
2003	15	18	2	2
2004. 1	25	18	1	-

Contracted services

by regional distribution:

	D	EU	nonEU
2000	16	5	1
2001	16	4	3
2002	12	6	9
2003	4	7	26
2004. 1	4	5	35

Why did vsatnet. com finally fail?

- GMD/ Fraunhofer merger changed the startup rules and the responsible persons
- In the years after Internet Hype Crash we couldnt find strong partners instead
- The Founders from science decided not to risk the retirement payments
- The risk of investing ca. 1. 0 Mio € too much for us
- We were no real entrepreneurs !

From Science to Business and back

- The scientific co-founders (the „good guys“) decided to keep their positions until retirement
- Our successors at Fraunhofer Research (the“bad guys“) were not able to „conquer“ the business for themselves, but forced us out of it
- Hence: We all remained caught in science, but
- I use this experience to teach students about business opportunities and failures

Место кибернетики и информатики в общем (школьном) образовании

Семенов А. Л., *чл.-корр. РАН*

ректор Московского института открытого образования

В докладе Виктора Васильевича Топоркова шла речь об истоках российской вычислительной техники и роли Московского энергетического института в ней. В связи с этим я так же хочу поделиться некоторыми персональными данными. Моя мать Евгения Тихоновна Семенова (урожденная Юркина) окончила МЭИ в марте 1950 г., то есть на год раньше Владимира Андреевича Мельникова (Мой отец – Лев Афанасьевич Семенов – конструктор магнетронов, также окончил МЭИ.) Е. Т. Семенова пришла по распределению в СКБ-245, в подразделение, которым руководил Башир Искандарович Рамеев. Ей, молодому специалисту, было поручено проектирование арифметического и логического устройства ламповой машины «Стрела». Е. Т. Семенова проработала в СКБ вплоть до запуска машины, а потом ушла оттуда в аспирантуру МЭИ, чтобы иметь больше времени для воспитания ребенка, которым и был я. Вся последующая работа Е. Т. Семеновой была связана с МЭИ, где она работала на Факультете автоматики и вычислительной техники вместе с упомянутыми В. В. Топорковым учеными, в том числе – и с Дмитрием Александровичем Поспеловым, который до сих пор остается другом нашей семьи.

Говоря о массовой образовательной информатике, часто упоминают стоявших у ее истоков четверть века назад А. П. Ершова и Е. П. Велихова. Я хотел бы упомянуть о роли В. А. Мельникова в критический для становления этой области период, совпавшей по времени с организацией Владимиром Андреевичем Института проблем кибернетики на базе Научного совета АН СССР по комплексной проблеме «Кибернетика». Владимир Андреевич сыграл решающую роль в становлении нового журнала «Информатика и образование» - сегодня лидера в своей области, имеющего значительный академический авторитет. Он же взял на себя заведование кафедрой Математической логики и теории алгоритмов мех-мата МГУ – после Андрея Николаевича Колмогорова, в далеко не безоблачное для кафедры время.

Говоря о содержании школьной информатики сегодня, можно начать с роли средств информационных и коммуникационных технологий (ИКТ) в школе. Сегодня в российской школе, как и в школе других стран мира, основным направлением развития является обеспечение доступа к средствам ИКТ (компьютерам, сканерам, принтерам, видеокамерам, цифровым измерительным приборам и т. д.) всегда, когда они могут быть эффективно использованы в образовательном процессе. Чтобы их использование было действительно эффективным, учащимся даются практические знания о технологии использования в контексте конкретных понятных и интересных учащемуся задач. Такое использование покрывает практически все школьные предметы. Параллельно с этим учащиеся осваивают и теоретическое, фундаментальное содержание информатики. При отборе основного ядра этого содержания важно выбирать систему понятий, утверждений, методов, которые, с одной стороны, действительно общезначимы, с другой – связаны с деятельностью учащегося по решению задач, достижению конкретных целей, прояснению наблюдаемых интересных явлений. Конечно, наряду с этим ядром курс информатики может (в не слишком большой доле) содержать и то, что называется «философские обобщения» и более специальные, в том числе – технологические вопросы, и захватывающие дух и воодушевляющие учащегося картины «современной информационной цивилизации» и важные для практики положения, относящиеся к технике безопасности, этике и закону в области информации. Однако сегодня основное содержание, «ядро» того или иного предмета будет определяться во многом тем, что входит в материал Государственной итоговой аттестации (за девятый и одиннадцатый классы школы). Недавно принятый российский закон фиксирует, что аттестация за 11 класс происходит в форме так называемого ЕГЭ (единого государственного экзамена). Группа математиков, информатиков, программистов, школьных учителей под руководством Льва Николаевича Королева

последние годы пытается проектировать содержание этой итоговой аттестации по информатике. Моя личная интерпретация современных представлений о фундаментальной информатике в школе содержится в следующем кратком (насколько это позволяет жанр доклада на юбилейной конференции) обзоре. Мне представляется, что эта интерпретация является естественным продолжением взглядов А. П. Ершова.

Начнем с того, что современная, подчеркнем, общекультурная, общечеловеческая грамотность предполагает понимание ряда понятий, которые сегодня можно отнести к области информатики. Они возникли и по сей день могут рассматриваться как входящие в понятийный аппарат таких областей как кибернетика, логика, общая теория систем, теория игр (сегодня есть несколько таких «теорий», относящихся к разным классам игр). К ним относятся такие понятия и их группы, как:

- Система, элементы и компоненты системы, их состояния, взаимодействие, сигналы
- Управление, обратная связь
- Игры (в том числе – с полной информацией), стратегии в играх

Еще раз отметим, что только что перечисленные понятия уже стали общекультурными. Школа должна на это реагировать. При этом мы должны дать учащимся задачи, где эти понятия работают. Можно строить простейшие конечные автоматы, можно программировать в визуальных языках движущихся по полу и управляемых роботов ЛЕГО (с микропроцессором, куда заложена составленная программа), можно играть в игры с камешками на столе или бросать кость и т. д.

Следующая область относится к современному понятию «being digital» (Н. Негропonte, МТИ), к «цифровому миру». Люди уже давно пользовались измерительными инструментами и сопоставляли с (мыслимой как непрерывная, аналоговая) длиной конечную десятичную дробь (целые: метры, сантиметры, миллиметры). Сегодня, чтобы обработать информацию о каких-то объектах или процессах, их смоделировать и спроектировать, нужно работать с их цифровыми, дискретными образами. Понятия перехода к такому образу – дискретизации (ср. «being digital») уже стало элементом общеинтеллектуального багажа современного человека. С этим связаны и классы дискретных, конечных объектов, изучаемых разделами современной математики, которые можно было бы назвать также информатической математикой или «mathematical computer science». В школе

(уже начальной) такие объекты могут быть наглядно и даже телесно представлены учащимся.

Это:

- цепочки (кортежи, конечные последовательности) бусин разной формы,
- те же бусины, сложенные в мешки (уже неупорядоченные множества объектов, среди которых могут встретиться и одинаковые),
- деревья,
- произвольные графы.

Последние типы объектов, конечно, полезны при наглядном рассмотрении упомянутых выше конечных автоматов и игр с полной информацией, а не наглядно их, до старшей школы, и не стоит рассматривать. Специалисту очевидно, что об упомянутых объектах можно формулировать много задач, в том числе и достаточно сложных. При этом замечательно, что эти задачи, в том числе и сложные, можно формулировать для учащихся даже начальной школы в наглядной форме, где объекты условия и решения задачи нарисованы на бумаге (или существуют в виртуальном микромире на экране компьютера). Одним из многих видов задач являются упражнения на кодирование разных математических объектов. Напомним, что мы начали с «дискретизации» образов реальных объектов. С этим связаны «прикладные» арифметические упражнения, относящиеся к тому, какой длины получится цепочка при дискретизации звука, изображения и т. д. В терминах цепочек можно говорить и о десятичных и двоичных числах и о хранении информации в разных видах памяти и об объеме информации (длине двоичной цепочки) и об объеме памяти, в которой могут храниться такие цепочки и о скорости передаче цепочек по каналу связи.

Задолго до появления компьютеров человек заинтересовался возможностью формального описания того, что можно называть «строгим рассуждением» и «верным высказыванием». Возникла логика, ставшая в период с 1900-ых по 1930-е гг. математической логикой. К тому же периоду относится и формализация того, что можно назвать «строгой, однозначно понимаемой инструкцией к действию». Возникла теория алгоритмов. Фундаментальные понятия и конструкции общей дисциплины «математическая логика и теория алгоритмов» были в середине XX века воплощены в «железе» и компьютерных кодах. (Здесь временное совпадение становится необыкновенно точным – в конце сороковых гг. и самом начале пятидесятых были созданы универсальные компьюте-

ры и А. А. Марковым было дано детальное доказательство правильности работы абстрактного универсального компьютера – в терминах структурного программирования, индуктивного доказательства с инвариантами и т. д.). Вот эти понятия:

- имена, выражения и их значения
- высказывания, их истинность, операции над высказываниями
- примеры правил логического вывода
- программа (запись алгоритма) и отвечающая ей последовательность действий (выполнение, вычисление)
- конструкции структурного программирования, переменные.

Упомянутые математические конструкции, относящиеся к формализованному мышлению и деятельности, сегодня постоянно применяются в информатике, относятся к ее фундаментальной, теоретической части. С другой стороны, они постепенно во все большей степени становятся частью общей культуры, не менее базовой, чем, скажем, арифметика процентов или вычисление площадей простейших геометрических фигур. При этом, как и в случае базовых объектов, упомянутых выше, освоение перечисленных понятий может идти уже в начальной школе на наглядном материале. Например, исполнение алгоритма может происходить в визуальном микромире на экране компьютера (команды выполняет робот в лабиринте или черепашка, перемещающаяся по плоскости и оставляющая след). Конкретные правила логического вывода также могут быть проиллюстрированы в визуальном контексте.

Следующий блок фундаментального информатического знания относится к базовым теоретическим достижениям в области основ информатики.

В него входят следующие темы:

- абстрактный компьютер (исполнитель) и язык программирования для него (имеется в виду какая-нибудь универсальная вычислительная модель)
- утверждение о существовании универсальной программы, позволяющей абстрактному компьютеру вычислить результат работы любой данной программы (в рассматриваемом языке) на любом исходном данном (это утверждение может быть строго доказано, что не входит в программу общеобразовательной школы, но пояснено на содержательном уровне оно может быть и в школе)
- тезис об универсальности данного компьютера и языка, состоящий в том, что с их помощью

можно вычислять значение любой вычислимой каким-либо образом функции (это утверждение относится к соотношению между формальным математическим объектом и интуитивным, неформальным с точки зрения математики, представлением; оно не может быть математически доказано и относится к информатике, но не к «чистой» математике).

Несмотря на естественность и важность двух последних утверждений, их введение даже в базовую подготовку программистов (не говоря уже о школьниках) может показаться не столь очевидным. Но в том-то и дело, что программист просто не задумывается о них, он программирует, их подразумевая. В общеобразовательном же смысле представление о том, что все можно вычислить на данном обозримом вычислительном устройстве, и что этому устройству можно «объяснить» любую вычислительную программу, очень важно. Задачи в данном блоке могут относиться к написанию простых программ для абстрактного компьютера, поясняющих его возможности по обработке цепочек или чисел. Представления о границах вычислимости, «алгоритмичности» и доказуемости естественно сочетаются с отрицательными утверждениями. Эти утверждения уже проникли в популярную математическую и околomатематическую литературу. К ним относятся:

- пример алгоритмической неразрешимости,
- принципиальная неполнота математических теорий.

«Диагональная» процедура, с помощью которой может быть доказана алгоритмическая неразрешимость, является одним из замечательных достижений математики. Замечательных в том числе и тем, что она доступна для понимания школьникам и не менее, чем иррациональность корня из двух, заслуживает присутствия в школьной программе.

Понятие сложности также относится к современному общекультурному багажу, оно же дает правильный взгляд на возможности информационных технологий. Это обуславливает полезность рассмотрения тем:

- сложность вычисления, как число его шагов,
- описательная сложность объекта, как длина кратчайшего описания,
- представление о случайности, как сложности

Выше уже упоминались задачи на построение алгоритмов. Безусловно, они являются важнейшей частью курса. Собственно теоретические

рассмотрения и определения не должны в нем занимать более 20–25% времени. К задачам относятся, прежде всего, анализ работы алгоритмов и построение алгоритмов – наиболее важная «задачная» часть курса. Трудность здесь состоит в выборе небольшого семейства важнейших конкретных алгоритмов. Вероятно, одним из подходов здесь может быть выбор алгоритмов, иллюстрирующих общие методы построения. Среди общих методов можно упомянуть (наряду со стилистикой структурного программирования, упомянутой выше):

- разбиение задачи на подзадачи, последующая сборка,
- переход от описания (например, системы функциональных уравнений) к вычислительному алгоритму. Среди конкретных алгоритмов:
 - *алгоритм перебора,*
 - *алгоритм сортировки.*

Здесь же естественно коснуться и темы:

- невозможность неперборных алгоритмов.

Описанный выше подход частично реализован в ряде школьных учебников для начальной и средней школы, написанных при участии докладчика, ряд тем отражены в учебниках, созданных в школах А. Г. Кушниренко и А. Шеня. Задачная его часть использовалась при разработке материалов государственной итоговой аттестации для школы. Для профильного (углубленного) изучения предназначен учебник: Информатика. Введение в компьютерные науки, Л. Н. Королева и А. И. Микова (<http://books.dore.ru/bs/flbid1600.html>).

Об идентификации нелинейных непрерывных систем

Шаншиашвили В. Г., *д. т. н., профессор*

Грузинский технический университет

Рассматриваются проблемы идентификации нелинейных непрерывных систем управления. Представлены методы структурной и параметрической идентификации нелинейных систем на множестве непрерывных блочно-ориентированных моделей, элементы которого – разные модификации моделей Винера и Гаммерштейна. Разработаны критерии определения структуры модели из множества моделей. Задача оценивания параметров сводится к решению систем алгебраических уравнений.

Применение современных методов управления требует наличия определенной информации о сущности процесса, протекающего в исследуемой системе, т. е. ее формализованного представления в виде уравнения, математической модели.

Одно из основных направлений в теории управления – идентификация систем предполагает построение модели на основании информации, полученной в условиях функционирования системы, с помощью наблюдений по ее входным и выходным переменным.

Идентификация систем основывается в основном на линейных стационарных моделях, которые широко применяются для производственных процессов. В то же время большинство реальных систем являются нелинейными и обладают, с точки зрения организации процесса управления, многими «неудобными» свойствами. При формализации закономерностей процессов, протекающих в нелинейных системах применение линейных или линеаризованных моделей возможно лишь в ограниченной области изменения переменных. Реальные системы могут быть адекватно охарактеризованы на всей рабочей области только нелинейными моделями.

При идентификации нелинейных систем нелинейными моделями можно выделить в основном две тенденции. Первая заключается в использовании моделей частного типа – в основном блочно-ориентированных моделей (например [1-5]), а вторая – общих моделей, в частности функциональных рядов Вольтерра (например [6-7]) и непрерывных и дискретных полиномов Колмогорова-Габова [8-10] для описания исследуемых систем. Существуют и другие подходы к решению задач идентификации нелинейных систем, напри-

мер разработка информационных и дисперсионных методов идентификации.

Использование как частных, так и общих моделей для идентификации нелинейных систем имеет свои положительные и отрицательные стороны. Преимущество частных моделей заключается в относительной легкости использования для представления нелинейных систем.

Идентификация систем связана с решением разных задач в зависимости от априорной информации о системе. Построение оптимальной модели системы во многом определяется решением задач структурной и параметрической идентификации.

Построение модели системы основывается на априорной информации о системе и апостериорной информации, полученной по результатам наблюдений над переменными входа и выхода системы. Задача определения структуры модели обычно решается на основе существующей априорной информации или использования физических законов процессов, протекающих в системе, по традиционной схеме идентификации [11]. Однако, определенная таким путем структура модели часто имеет высокую размерность, и поэтому остается довериться идентификации и оцениванию.

Подход при решении задачи определения структуры модели нелинейных систем, на основе использования апостериорной информации, делается обычно из позиции упомянутых выше двух тенденций.

При представлении нелинейных систем блочно-ориентированными моделями основные результаты в сфере структурной идентификации получены при идентификации дискретными моде-

лями, в составе которых нелинейное звено обычно описывается полиномиальной функцией второй степени, а линейные звенья – разностными уравнениями (например [3]).

При идентификации нелинейных систем общими моделями также возникает задача выбора оптимальной структуры из общей модели или статистически значимых членов, которая решается обычно на основе информационных критериев Фишера, Акаике, Бейеса и др. (например [10]).

В данной работе задачи структурной и параметрической идентификации нелинейных непрерывных систем рассматривается на множестве непрерывных блочно-ориентированных моделей, элементы которого – разные модификации моделей Винера и Гаммерштейна (рис. 1).

В общем случае нелинейное статическое звено, входящее в состав моделей, описывается уравнением

$$y(t) = f(u) = \sum_{i=1}^n c_i u^i(t), \quad (1)$$

где $u(t)$ и $y(t)$ – входная и выходная переменные, соответственно, $c_i (i = 0, 1, \dots, n)$ – постоянные коэффициенты. Входящие в состав блочно-ориентированных моделей линейные динамические звенья, с передаточными функциями в операторной форме $W(p)$, $W_i(p)$ ($i = 1, 2, 3, 4$), устойчивы.

Задача структурной идентификации ставится на основании классического определения идентификации Л. Заде, т. е. заданы классы моделей и входных сигналов, требуется разработать критерии идентификации структуры модели на множестве моделей.

Представлен метод структурной идентификации в установившемся режиме при входных периодических воздействиях, имеющих равномерно и абсолютно сходящиеся ряды Фурье. По критерию идентификации структуры модели, разработанному по зависимости постоянной составляющей этих колебаний от периода входного сигнала, для представления структуры модели исследуемой нелинейной системы, можно выделить три подмножества множества моделей, элементы которых: 1) простая и обобщенная модели Гаммерштейна; 2) простые и обобщенные модели Винера и Винера-Гаммерштейна и 3) расширенные модели Винера и Винера-Гаммерштейна.

При определении структуры модели возможно определение принадлежности структуры модели к определенным подмножествам множества моделей.

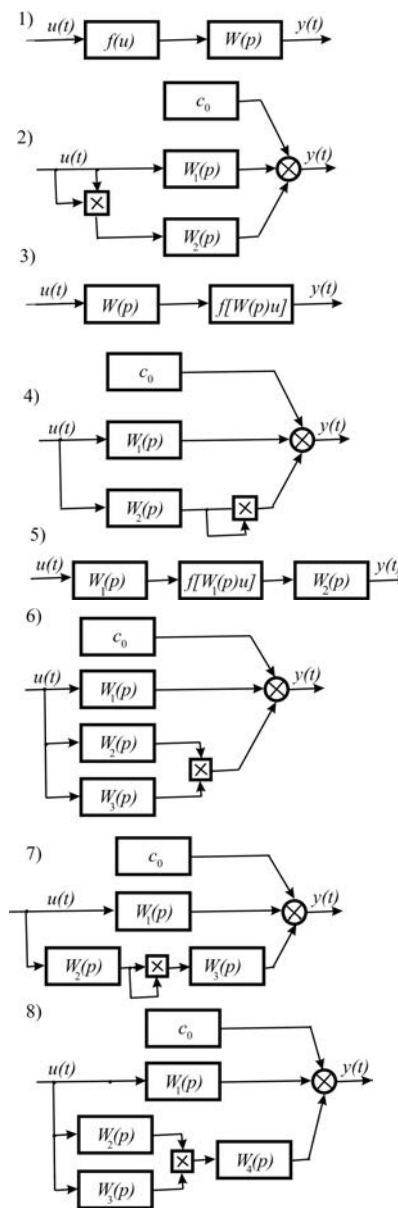


Рис. 1. Блочно-ориентированные модели:

- 1) простая модель Гаммерштейна;
- 2) обобщенная модель Гаммерштейна;
- 3) простая модель Винера;
- 4) обобщенная модель Винера;
- 5) простая каскадная модель Винера-Гаммерштейна;
- 6) расширенная модель Винера;
- 7) обобщенная каскадная модель Винера-Гаммерштейна;
- 8) расширенная каскадная модель Винера-Гаммерштейна.

С целью определения структуры модели из каждого подмножества, в работе представлен метод структурной идентификации при воздействии на входе системы синусоидального сигнала:

$$u(t) = U \cos \omega t. \quad (2)$$

По критерию структурной идентификации, разработанному по зависимости их постоянной составляющей от частоты входного сигнала и между постоянной составляющей, первой и второй гармоники, возможно определение каждой структуры из множества моделей.

Методы структурной идентификации при периодических и синусоидальных воздействиях подразумевают проведение активного эксперимента и подачу специальных тестовых сигналов на вход исследуемой системы, что для ряда практических систем не является возможным.

В работе представлен метод структурной идентификации нелинейных систем, который не требует изменения условий нормального функционирования системы. В этом случае подразумевается то, что входная переменная системы – стационарный центрированный случайный процесс с дисперсией σ^2 , коэффициентом корреляции $R = R(\tau)$ и распределением:

$$w(u) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{u^2}{2\sigma^2}}. \quad (3)$$

По критерию идентификации структуры модели, разработанному по зависимости постоянной составляющей корреляционной функции выходного процесса от изменения спектра частот входного случайного процесса, возможно выделение из множества моделей двух подмножеств, элементы которых: 1) модели Гаммерштейна; 2) модели Винера и Винера-Гаммерштейна для представления структуры модели исследуемой системы.

Что касается оценивания параметров блочно-ориентированных моделей, то это довольно сложная задача, из-за того что большинство этих моделей, кроме моделей Гаммерштейна (простой и обобщенной) являются нелинейными относительно параметров, а также из-за большого числа оцениваемых параметров. Так например, число оцениваемых параметров простой каскадной модели Винера-Гаммерштейна, линейные динамические звенья которой описываются дифференциальными уравнениями m_1 и m_2 порядка, равно $n + m_1 + m_2 + 3$. Поэтому решение задачи параметрической идентификации в аналитическом виде возможно только для некоторых блочно-ориентированных моделей низкого порядка.

В данной работе рассматривается задача параметрической идентификации нелинейных систем на основе активного и пассивного эксперимента, которая непосредственно связана с задачей структурной идентификации с использованием выражений и экспериментальных данных, полученных для последней. С помощью аппроксимации Фурье и экспонен-

циальной аппроксимации задача оценивания параметров сводится к решению систем алгебраических уравнений.

Литература

1. Haber R., Keviczky L. Identification of nonlinear dynamic systems // Preprints of the IV IFAC Symposium on Identification and System Parameter Estimation. M.: Institute of Control Sciences, 1976. Part 1. P. 62-112.
2. Billings S. A., Fakhouri S. Y. Identification of systems containing linear dynamic and static nonlinear elements // Automatica. 1982. Vol. 18, No. 1. P. 15-26.
3. Haber R., Unbehauen H. Structure identification of nonlinear dynamic systems – a survey on input/output approaches // Automatica. 1990. Vol. 26, No. 4. P. 651-667.
4. Shanshiashvili B. G. Frequency method for identification of a model structure of nonlinear continuous – time systems // Preprints of the 9 th IFAC/IFORS Symposium on Identification and System Parameter Estimation. Budapest.: Pergamon Press, 1991. Vol. 1. P. 640 – 643.
5. Шаншиашвили В. Г. Структурная и параметрическая идентификация определенного класса нелинейных систем в частотной области. // Труды V международной конференции “Идентификация систем и задачи управления” SICPRO '06. М., ИСУ РАН, 2006. С. 189-202
6. Пупков К. А., Капалин В. И., Ющенко А. С. Функциональные ряды в теории нелинейных систем. М.: Наука, 1976. 448 С.
7. Parker G. A., Moore E. L. Practical nonlinear system identification using a modified Volterra series approach // Automatica. 1982. Vol. 18, No. 1. P. 85-91.
8. Колмогоров А. Н. Интерполирование и экстраполирование стационарных случайных последовательностей // Изв. АН СССР, серия математическая. 1941. т. 5, № 1. С. 3-14.
9. Gabor D., Wilby P. L., Woodcock R. A universal nonlinear filter predictor and simulator which optimizes itself by a learning process // IEE Proc. 1961. Vol. 108, Part B. P. 422-433.
10. Kortmann M., Unbehauen H. Structure detection in the identification of nonlinear systems // APII. Vol. 22, No. 1. P. 5-25.
11. Эйкхофф П. Основы идентификации систем управления. М.: Мир, 1975. 683 с.

Моделирование тепловых режимов в электронных модулях

Мадера А. Г., *д. т. н., профессор*
Кандалов П. И.

НИИ Системных исследований РАН

Статья посвящена изложению метода математического и теплового моделирования трехмерных температурных полей в сложных конструкциях реальных электронных модулей, содержащих многослойную печатную плату и установленные на ней интегральные микросхемы и электро- радио-элементы. Метод моделирования реализован в виде программного пакета для персональных компьютеров, доведенного до уровня пользователя. Рассмотрены результаты моделирования теплового режима реальной конструкции электронного модуля.

Электронный модуль (ЭМ) является важнейшей структурной единицей любого электронного устройства и представляет собой систему, состоящую из многослойной печатной платы (МПП) и установленных на ней микросхем (МС) и электро-радиоэлементов (ЭРЭ) – диодов, транзисторов, резисторов, конденсаторов, катушек индуктивности, электрических разъемов и т. д. Проектирование ЭМ включает в себя решение многочисленных проблем, таких, например, как разработка логической схемы ЭМ, разработка принципиальной электрической схемы, трассировка сигнальных проводников в слоях МПП, выбор и/или конструирование МС, корпусов МС, компоновка МС и ЭРЭ на плате и многие другие. Результаты многочисленных исследований показали, что только 5-10% потребляемой электронной системой мощности превращается в полезные сигналы, остальные же 90-95% потребляемой мощности переходят в тепловую энергию, которая рассеивается в конструкции системы приводя к нагреванию всех элементов, установленных в ЭМ. Поскольку значения мощностей потребляемых микросхемами и ЭРЭ различны и все они находятся в тепловом взаимодействии между собой и МПП, то во всей конструкции ЭМ и в ее элементах возникает неоднородное температурное поле.

Из-за температурной зависимости электрических параметров МС и ряда ЭРЭ, возникающее в конструкции ЭМ температурное поле приводит к тому, что важнейшие электрические характеристики ЭМ, такие как статические и динамические электрические параметры, надежность, быстродействие, помехоустойчивость и т. д. претерпева-

ют существенные изменения. Перегревы элементов ЭМ могут приводить к выходу ЭМ из строя, неправильному срабатыванию, низкой надежности и т. д. Поэтому при конструировании и проектировании ЭМ фактор возникновения температурного поля относится к числу нежелательных, воздействие которого стремятся, по возможности, свести к минимуму. С этой целью предпринимаются меры различного характера, которые уменьшают уровни температуры на кристаллах МС и других ЭРЭ, что достигается многочисленными конструктивными и схемотехническими способами, например,

- снижением уровней рассеиваемых в МС мощностей,
- использованием корпусов МС с низкими значениями тепловых сопротивлений,
- установкой локальных теплоотводов (радиаторов) на корпусах МС, имеющих значительную потребляемую мощность,
- использованием методов интенсификации теплообмена (высокотеплопроводные шины, обдув вентиляторами, жидкостное охлаждение, тепловые трубы и т. д.).

Конструкции ЭМ могут эксплуатироваться в различных, в том числе и аномальных условиях, при воздействии экстремальных температур среды, влажности, высоких уровнях радиации, повышенных тепловых нагрузках и, кроме того, в жестко ограниченных по размерам малых объемах, размеры которых затрудняют физический доступ к тепловыделяющим элементам и не позволяют осуществлять эффективный отвод тепловой энергии.

Для оптимальной организации эффективного отвода тепловой энергии в конструкции ЭМ и обеспечения требуемого уровня температур необходима тщательная проработка и решение большого числа самых разнообразных проблем, которые решаются в процессе теплового проектирования ЭМ. Тепловое проектирование включает в себя решение следующих проблем и задач:

- моделирование температурных полей элементов и конструкции ЭМ при различных условиях эксплуатации и испытаний;
- проектирование элементов и систем отвода теплоты (конвективно-воздушного, кондуктивно-воздушного, кондуктивно-жидкостного, кондуктивно-испарительного, жидкостного и т. д.);
- конструирование системы эффективного охлаждения элементов ЭМ и всей конструкции ЭМ в целом.

Моделирование тепловых процессов в ЭМ представляет собой комплексную проблему и включает в себя как математическое, так и компьютерное моделирование. Сложность моделирования температурных полей ЭМ проявляется уже в том, что на сегодняшний день, отсутствуют надежные, адекватные и универсальные методы компьютерного моделирования тепловых процессов сложных конструкций ЭМ. Так, известный на Западе программный комплекс Beta-Soft позволяет моделировать температурные поля в ЭМ только для печатных плат, имеющих три слоя, а отечественный программный комплекс АСОНИКА – только один слой. Между тем, современные МПП насчитывают десятки слоев, каждый из которых имеет различные толщину и теплофизические параметры.

При математическом и компьютерном моделировании тепловых процессов необходимо учитывать конструктивно-физические особенности ЭМ, такие как:

- конструкция ЭМ представляет собой сложную систему, состоящую из большого числа элементов, многие из которых сами, в свою очередь, являются сложными системами. Так, микросхемы, входящие в состав ЭМ, представляют собой конструкцию, состоящую из кристалла с созданной на его поверхности системой $p-n$ -переходов, металлизированной разводки, корпуса МС, выводов кристалла и корпуса, элементов теплоотвода и охлаждения;
- элементы ЭМ характеризуются различными значениями потребляемых мощностей, размерами и пространственной конфигурацией, различными теплофизическими характеристиками материала,

неоднородными включениями и полостями, нестационарным распределением потребляемых элементами мощностей как во времени, так и в пространстве и т. д. ;

- печатные платы и МС, определяющие работоспособность и эксплуатационные характеристики ЭМ, представляют собой многослойные структуры, состоящие из большого числа слоев различной толщины и выполненных из разнородных материалов.

Неравномерное распределение МС, установленных на МПП, многообразие их конструктивно-исполнения, различные значения потребляемых мощностей приводит к тому, что температурные поля ЭМ носят существенно трехмерный и неоднородный характер. Кроме того, уравнения математической модели в общем случае являются также и нелинейными, что обуславливается нелинейным характером температурных зависимостей процессов теплообмена и потребляемых МС мощностей.

В данной статье рассматривается метод математического и компьютерного моделирования температурных полей конструкций ЭМ. Методы позволяют проводить расчеты, с наперед заданной точностью, трехмерных температурных полей сложных конструкций ЭМ, содержащих МПП и установленные на ее обеих поверхностях МС и ЭРЭ. Методы моделирования учитывают реальные конструктивные особенности корпусов МС, условия их монтажа на поверхности МПП, особенности крепления МПП в ЭМ, блоках и панелях, конструкции отвода теплоты от элементов ЭМ. Компьютерное моделирование, реализующее математическое моделирование, реализовано в виде программного комплекса написанного на языке Delphi для персональных компьютеров и не имеет аналогов в российской и зарубежной практике моделирования ЭМ. В отличие от существующих программных комплексов моделирования температурных полей в ЭМ (Beta-Soft, АСОНИКА и др.), количество слоев в МПП, количество МС и ЭРЭ не ограничены.

Рассмотрим тепловую модель ЭМ (рис. 1). Элементы ЭМ, установленные на МПП (МС, ЭРЭ, электрические разъемы), подразделяются на два вида. Первый вид элементов – *активные элементы*, – характеризуется определенным уровнем потребляемой мощности, которая приводит к активному внутреннему тепловыделению. К активным элементам относятся: МС, диоды, транзисторы, резисторы, катушки индуктивности, трансформаторы. Ко второму виду элементов ЭМ отно-

сятся *пассивные элементы*, которые не обладают собственным тепловыделением, это – электрические разъемы, конденсаторы, теплоотводы (радиаторы, метал-лические детали, тепловые трубы). Посредством конструктивного исполнения активных и пассивных элементов создается надежное крепление элементов к МПП и тем самым обеспечивается эффективная тепловая связь

элементов с МПП и через МПП – друг с другом. При тепловом моделировании пассивные элементы выполняют функции теплоотводов, и по своему физическому смыслу являются стоками теплоты, отводимой от МПП, а активные элементы, выделяющие тепловую энергию, играют в конструкции ЭМ роль источников теплоты.

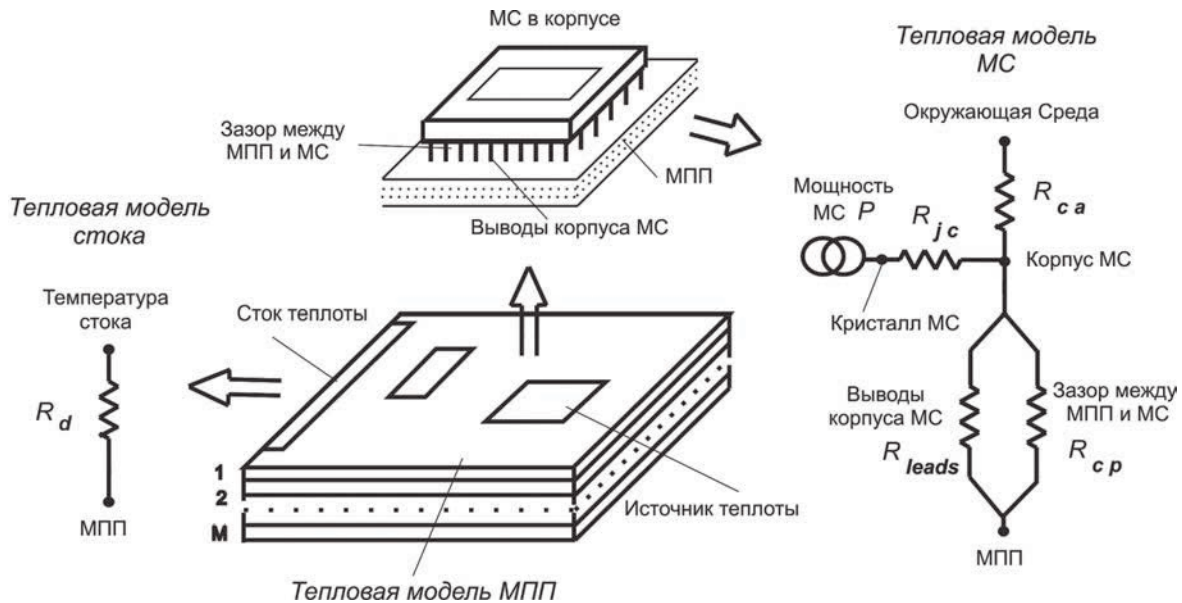


Рис. 1. Тепловые модели электронного модуля: МПП, МС, ЭРЭ, электрических разъемов и других элементов ЭМ

Активные и пассивные элементы моделируются с мощью тепловых схем, которые получают методом электротепловой аналогии на макроуровне моделирования [1]. Согласно этому методу, сложная конструктивная система, для которой осуществляется моделирование теплового режима, разбивается на систему состоящую из нескольких изотермических тел, связанных в тепловом отношении между собой и с окружающей средой. Разбиение на систему тел производится таким образом, что с достаточной для практики точностью, можно принять, что каждое тело системы имеет однородное температурное поле, характеризующее средней температурой. Поэтому каждое тело системы может быть смоделировано в виде узла, имеющего среднюю температуру, а тепловое взаимодействие тел – тепловыми сопротивлениями, которые входят в ветви, соединяющие узлы, соответствующие взаимодействующим телам. Источники тепловой энергии, выделяемой активными элементами в результате собственного энергопотребления, моделируются независимыми источниками потока, а заданные температуры среды – узлами с заданными температурами. В

результате электротепловой аналогии получается тепловая схема, состоящая из узлов, соединяющих их ветвей с включенными тепловыми сопротивлениями, и источников потока или температуры. Аналогия между электрической схемой и ее электрическими характеристиками и между тепловой схемой и ее тепловыми характеристиками, проявляется в следующем соответствии: электрический ток – тепловой поток, потенциал узла – температура тела, электрический ток в ветви – тепловой поток в ветви, независимый источник тока – источник теплового потока, источник потенциала – заданная температура среды, электрическое сопротивление – тепловое сопротивление. Тепловые схемы активных и пассивных элементов рассмотрены ниже.

Тепловая модель МПП представляет собой многослойную структуру (рис. 1), которая состоит из множества тонких разнородных слоев, различной толщины, количество которых достигает несколько десятков. С верхней и нижней поверхностей многослойной структуры МПП происходит конвективный теплообмен в среду по закону Ньютона. Слои в многослойной структуре МПП

могут быть четырех видов: диэлектрические слои из полиимида, тонкие медные энергетические слои для земли и питания, сигнальные слои с металлическими сигнальными проводниками прямоугольного сечения, теплоотводящие сравнительно толстые медные слои. Помимо слоев, выполненных из однородных материалов, реальная конструкция МПП включает в себя также сигнальные слои, имеющие композитную структуру, и которые состоят из сигнальных проводников прямоугольного сечения, выполненных, как правило, из меди, и разделяющих их диэлектрических слоев, изготавливаемых, как правило, из полиимида. Поскольку расположение медных сигнальных проводников в слое диэлектрика соответствует маршрутам их трассировки и отличается от слоя к слою, то теплопроводность такого композитного слоя будет анизотропной и различаться в каждой точке слоя. Однако для того, чтобы моделирование температурных полей слоев МПП было возможным, необходимо от анизотропной теплопроводности слоя уметь переходить к некоторой ее эффективной усредненной величине $\lambda_{эфф}$. Определение эффективной теплопроводности $\lambda_{эфф}$ каждого слоя представляет собой довольно сложную проблему, не решенную до сих пор. В ряде случаев, расчет эффективной теплопроводности $\lambda_{эфф}$ тепловой модели МПП с достаточной для практики моделирования погрешностью, можно рассчитывать по формуле ожидаемой или взвешенной величины $\lambda_{эфф} = \alpha_1 \lambda_1 + \alpha_2 \lambda_2$, $\alpha_1 + \alpha_2 = 1$, где λ_1 и λ_2 – теплопроводности металлического проводника и диэлектрика, занимающих в объеме композитного слоя доли, равные α_1 и α_2 соответственно.

На верхней и нижней сторонах тепловой модели МПП расположены прямоугольные площадки, которые соответствуют двум видам элементов ЭМ: активным элементам, которые выделяют тепловую энергию и пассивным элементам – не обладающим этим свойством. Размеры прямоугольных площадок на поверхностях МПП вдоль осей X и Y равны соответствующим размерам проекции элементов на поверхности МПП. Площадки, которые соответствуют активным элементам играют для МПП роль источников тепловой энергии, поступающей от элемента ЭМ в МПП; будем называть их источниками теплоты. Площадки, которые соответствуют пассивным элементам играют роль стоков тепловой энергии, отводимой от МПП; будем называть их стоками теплоты. С достаточной для инженерной практики точностью моделирования можно считать, что теплоотдачей с торцев МПП можно пренебречь по срав-

нению с теплоотдачей с верхней и нижней поверхностей МПП и считать, что торцы прямоугольного параллелепипеда в тепловой модели МПП теплоизолированы. Для общности моделирования принято, что температуры среды и условия теплообмена на верхней и нижней поверхностях МПП могут быть различными. Зависимостью теплофизических параметров материалов тепловой модели, а также мощностей активных элементов ЭМ от температуры пренебрегается.

Математическая модель описывающая трехмерное стационарное температурное поле $T_i = T_i(x, y, z)$ в каждом i -м слое тепловой модели МПП представляет собой систему следующих уравнений:

- уравнение теплопроводности в i -ом слое

$$\frac{\partial^2 T_i}{\partial x^2} + \frac{\partial^2 T_i}{\partial y^2} + \frac{\partial^2 T_i}{\partial z^2} = 0,$$

$$i = 1, 2, \dots, M$$

- граничные условия на верхней ($i = 1$) и нижней ($i = M$) поверхностях тепловой модели МПП, с которых происходит теплообмен в среду по закону Ньютона и одновременно – поступление тепловой энергии через источники теплоты в МПП и отвод тепловой энергии от стоков теплоты, расположенных на поверхностях МПП, суммарная мощность которых равна P_1 для $i = 1$ (верхний слой МПП) и P_M – для $i = M$ (нижний слой МПП)

$$\lambda_1 \frac{\partial T_1}{\partial z} \Big|_{z=d_1} = -\alpha_1 (T_1 \Big|_{z=d_1} - T_{a1}) + P_1,$$

$$\lambda_M \frac{\partial T_M}{\partial z} \Big|_{z=0} = \alpha_M (T_M \Big|_{z=0} - T_{a2}) - P_M$$

$$P_1 = \sum_{j=1}^k \frac{P_{j1} h_{j1}}{l_{x_{j1}} l_{y_{j1}}}, \quad P_M = \sum_{j=1}^l \frac{P_{jM} h_{jM}}{l_{x_{jM}} l_{y_{jM}}}$$

- граничные условия на границе i -го и $i+1$ -го слоев тепловой модели МПП, выражающие равенство температур и потоков теплоты на стыке двух соседних слоев:

$$T_i \Big|_{z=0} = T_{i+1} \Big|_{z=d_{i+1}},$$

$$\lambda_i \left. \frac{\partial T_i}{\partial z} \right|_{z=0} = \lambda_{i+1} \left. \frac{\partial T_{i+1}}{\partial z} \right|_{z=d_{i+1}},$$

где P_{j1}, P_{jM} – мощности j -го источника (s) или стока (d) теплоты на верхней и нижней поверхностях тепловой модели МПП, с размерами по осям X и Y равными l_{xj1}, l_{yj1} и l_{xjM}, l_{yjM} соответственно; α_1, α_M и T_{a1}, T_{a2} – коэффициенты теплоотдачи и температуры среды с верхней и нижней поверхностей тепловой модели МПП; h_j – функция Хевисайда равная 1, если точка на поверхностях тепловой модели МПП с координатами (x_j, y_j) принадлежит источнику теплоты и равна 0 – в противном случае, т. е. $h_j = 1$, если $(x_j, y_j) \in P_j$ и $h_j = 0$, если $(x_j, y_j) \notin P_j$

Математическая модель является линейной, так как в тепловой модели принято, что мощности источников теплоты, а также теплофизические параметры материалов тепловой модели, не зависят от температуры и координат.

Граничные условия моделируют конвективный теплообмен с внешних поверхностей тепловой модели МПП в среду по линейному закону Ньютона. В то же время необходимо понимать, что в реальности конвективный теплообмен со средой в общем случае представляет собой сложный нелинейный процесс взаимодействия потока жидкости (газа) с нагретой поверхностью, описываемый системой уравнений Навье-Стокса, решение которой не получено до сих пор. Для практических задач теплообмена, в которых рабочие температуры ЭМ не допускаются выше 120°C , моделирование теплообмена по линейному закону Ньютона является достаточно адекватным и приводит к результатам имеющим достаточную степень точности. В линейном законе Ньютона вся сложность теплообмена со средой сосредоточена в коэффициенте теплоотдачи α . Коэффициенты теплоотдачи вообще говоря нелинейно зависят от температуры поверхности и температуры среды и вычисляются по эмпирическим формулам, которые получают на основании экспериментальных данных с их дальнейшей математической обработкой [2, 3].

В статье разработан новый метод решения уравнений математической модели, который эффективно достигает цели как для плохо обусловленных матриц, так и для матриц, норма которых больше единицы. В первом случае обычные методы не позволяют найти решение уравнений из-за проблем с устойчивостью и расходимостью, а во втором – используемые итерационные методы расходятся. Предложенный численный метод решения уравнений математической модели основан на идее последовательных приближений и использует комбинацию методов релаксации [5] и половинного деления [6]. Моделирование, проведенное для различных конструкций ЭМ, с помощью разработанного метода, показало его эффективность, быструю сходимость к искомому решению (в ряде случаев за 3 – 5 итераций), а также независимость метода от меры обусловленности матрицы системы уравнений математической модели.

С помощью разработанных методов было проведено моделирование температурных полей различных конструкций ЭМ, показавшее хорошее согласие с данными эксперимента.

Литература

1. Мадера А. Г. Моделирование теплообмена в технических системах. – М. : НФ, 2005
2. Дульнев Г. Н. Тепло- и массообмен в радиоэлектронной аппаратуре. – М. : Высш. шк. , 1984
3. Справочник по теплообменникам: В 2 т. / Пер. с англ. , под ред. Б. С. Петухова. – М. Энергоатомиздат, 1987
4. Кандалов П. И. , Мадера А. Г. Моделирование температурных полей в многослойных структурах // Программные продукты и системы, №4 (84), 2008. С. 46 – 49
5. Фаддеев Д. К. , Фаддеева В. Н. Вычислительные методы линейной алгебры. – М. : Физматгиз, 1960.
6. Демидович Б. П. , Марон И. А. Основы вычислительной математики. – М. : Наука, 1966.

Электричество и данные в одном потоке

Сургуладзе М. Ш., доктор инфотелекоммуникаций

НИИСИ РАН, *sfm12@mail.ru*

Мегабайты по электропроводке

Несмотря на относительную новизну и неширокое распространение технологии сетей передачи данных по электропроводке, события в этом сегменте все же происходят. Ведущие производители оборудования и поставщики электроэнергии вновь стали проявлять интерес к технологии Power Line Communication.

Бытующее мнение, что большинство изобретений происходят от человеческой лени, очень хорошо применимо к сетям передачи данных. С тех пор как связь между компьютерами в офисе стала обязательной, не прекращались попытки максимально упростить ее – например, избавиться от необходимости прокладывания дополнительных кабелей. Сейчас разрабатываются все новые технологии, которые позволяют «сэкономить» на разводке сетевой инфраструктуры отдельного здания (HomePNA, предполагающие использование телефонной проводки, или беспроводные сети Wi-Fi). Технология, о которой пойдет речь, использует для построения локальных сетей обычную электропроводку, существующую в любом здании. Называется она Power Line Communication (PLC), или PowerLine.

У PowerLine – достаточно сложная и переменчивая судьба. Несколько раз ей прочили центральное место в развитии домашних информационных сетей. Потом «забывали» о ней, чтобы по мере совершенствования технологической базы вновь вернуться и провозгласить ее чуть ли не панацеей. Развитие PLC – это череда тактических исследовательских побед и стратегических рыночных поражений. PowerLine – новшество только на первый взгляд: высокочастотную связь по линиям электропередач в мире применяют с 20-х годов XX века, в основном для организации канала связи в энергосистеме (речь идет о высоковольтных линиях).

Эксперименты в области передачи данных по электросети велись достаточно давно, но низкая скорость и слабая помехозащищенность были наиболее узким местом данной технологии. В то время PLC-системы использовались в основном

для сигнализации в энергосистемах и на железных дорогах

Период «отрочества» в развитии технологии связан с проводимыми в 1997–2000 годах экспериментами по передаче данных и голоса в рамках пилотных проектов ведущих исследовательских лабораторий. Кроме малоизвестных на тот момент компаний, в них принимали участие такие гиганты телекоммуникационной индустрии, как Siemens, Nortel и несколько провайдеров услуг связи Германии и Великобритании. И хотя планы строились поистине наполеоновские (проект компании Norweb Telecom, объединивший усилия канадской Nortel Networks и английской United Utilities) и обещания раздавались щедро (1,5 пфеннига за минуту работы в Интернете), идея широкомасштабного использования электропроводки «не повезло».

Лишь появление более мощных DSP-процессоров (цифровые сигнальные процессоры) и использование таких способов модуляции сигнала, как OFDM-модуляция (Orthogonal Frequency Division Multiplexing), позволили значительно продвинуться вперед в реализации технологии PLC.

Несколько лет назад лидеры создали телеком-альянс под названием HomePlug Alliance. Компании-члены занимаются совместным проведением научных исследований и практических испытаний, а также принятием единого стандарта на передачу данных по системам электропитания. Прототипом PowerLine является технология PowerPacket фирмы Intellon, положенная в основу создания единого стандарта HomePlug1.0 specification (принятого альянсом HomePlug в 2001 г.), в котором определена скорость передачи данных до 14 Мбит/с.

Смещение интересов

Первоначально все усилия по созданию PLC-систем были направлены на решение проблемы «последней мили». Считалось, что новая технология составит достойную конкуренцию своим именитым собратьям – ADSL и кабельным модемам.

Однако в процессе разработки реального оборудования возник целый ряд технических и нормативно-правовых препятствий. Самые серьезные из них – борьба с помехами и строгое соблюдение требований электромагнитной совместимости (ЭМС) с радиоловительскими службами, работающими в диапазоне частот 1,6–30 МГц. Именно последний фактор сыграл отрицательную роль в распространении PLC на территории европейских стран. В результате технологическая новинка «переехала» в Японию, Бразилию, Индию, Россию, где в последнее время работа по ее использованию весьма активизировалась. И хотя в Европе интерес к данной технологии отнюдь не угас, акцент заметно сместился с подключения к Интернету на организацию домашних сетей.

Дом, полный электроприборов и радушно обслуживающий своих хозяев, появился в фантастических романах Рэя Брэдбери еще в 60-х годах. Сегодня эта идея уже не воспринимается как совершенно фантастическая, однако ее реализация отнюдь не дешевая и далеко не жизненно необходимая. Но ведь в исходной идее – связать воедино контроллер (блок управления), компьютер, принтер, телефон, датчики климат-контроля и различные исполнительные устройства (управляемые выключатели, кондиционер, обогреватели, кухонные бытовые приборы, аквариум и систему полива газона) – ничего сверхъестественного нет. Очевидно, что такая единая среда, включающая и ПК, и компьютерную периферию, и бытовые устройства, сможет функционировать только при наличии локальной информационной сети. Основное требование к ней – высокий и гарантированный уровень надежности, обусловливаемый в первую очередь степенью совершенства используемой технологии обмена данными. Рынок домашних сетей зародился всего несколько лет назад и своим «происхождением» во многом обязан Интернету, поскольку первые подобные сети создавались в первую очередь ради того, чтобы через сеть получить доступ к интерактивным играм и развлечениям. Сегодня ситуация радикальным образом изменилась, и рынок домашних сетей развивается очень высокими темпами. Оказалось, что при большом количестве пользователей домашней сети легче и, главное, экономически выгоднее организовать широкополосный доступ для всех, чем тянуть индивидуальные линии для каждого. И здесь PLC-технологии могут продемонстрировать свои низкие начальные капиталовложения; быстрое развертывание и возможность поэтапного наращивания по мере необходимости; обеспечение не только высокоскоростного

доступа в Интернет, но и телефонной связью (локальной, с выходом в городские телефонные сети); комплексное предоставление энергетических и телекоммуникационных услуг одним поставщиком. Кстати, именно с появлением новых версий технологий передачи информации связывается процесс переосмысления концепции домашней системы безопасности, включающей широкий спектр датчиков (пожарных, движения, разбития стекла и др.), подсистем мониторинга (в том числе камер слежения), средств пожаротушения и управления доступом к объектам. Операторам энергокоммуникаций новая технология также предоставляет широкие возможности по контролю за оборудованием, дистанционной диагностике и управлению. Кроме того, значительно упрощается процесс тарификации энергопотребления.

Суть технологии

Технология PowerLine основана на частотном разделении сигнала, передаваемого по силовым кабельным линиям, когда высокоскоростной поток данных разбивается на несколько низкоскоростных, каждый из которых передается на отдельной частоте с последующим их объединением в один сигнал. Использование 84 так называемых поднесущих частот в диапазоне 4–21 МГц не оказывает влияния на передачу по проводам обычной электроэнергии, поскольку составляет огромную разницу в сравнении со стандартными 50 или 60 Гц электрической цепи. Таким образом, «обычная» электросеть может одновременно доставлять электроэнергию и данные по одной цепи (линии). При этом PLC-устройства могут «видеть» и декодировать информацию, хотя обычные электрические устройства – лампы накаливания, двигатели – даже «не догадываются» о присутствии сигналов сетевого трафика и работают в обычном режиме.

Однако реализация систем передачи данных по электрическим линиям в России связана с дополнительными трудностями. Прежде всего, по техническим характеристикам отечественные электрические сети отличаются от западных, кроме того, и это главное, отсутствуют стандарты, определяющие основные параметры систем передачи данных по линиям электропитания.

При передаче сигналов по обычной «бытовой» электросети могут возникать большие затухания сигнала на определенных частотах, что может привести к потере данных. В технологии PowerLine предусмотрен способ решения этой проблемы – динамическое включение и выключение передачи сигнала (Dynamically turning off and

on data-carrying signals). Специализированное устройство постоянно отслеживает состояние канала передачи информации и, «заметив» участок спектра с превышением определенного порогового значения затухания, прекращает использование этих частот до восстановления нормального значения затухания. Технология PLC позволяет также организовать «удлинитель» телефонной линии все по той же электросети. С помощью специального адаптера телефон можно перенести на 120–150 м. На Западе эта техника для предоставления скоростного доступа в Интернет.

Вопросы безопасности

При передаче информации в PLC-сетях, как и любых других сетях общего доступа, актуальной является проблема ее защиты. Чтобы не допустить несанкционированного доступа к информации в процессе ее передачи, большинство сетевых PLC-адаптеров оборудовано средствами шифрования сигнала «на лету» (как правило, 56-битовой технологией DES). Кроме того, устройства электрических сетей могут разрешать доступ только тем пользователям, чьи серийные номера, номера карт доступа к каналу связи или другие уникальные идентификаторы известны сетевому администратору. Перечисленные возможности могут быть реализованы в дополнение к различным внутренним механизмам безопасности – паролированию, программным правам доступа к ресурсу и т. п. – встроенным в сеть. В конечном результате достигается как минимум столь же высокий уровень безопасности, как в большинстве беспроводных сетей.

PLC в мире и России

Еще в июле 2001 г. немецкая компания RWE организовала на базе оборудования Ascom сеть PowerLine, предоставлявшую доступ в Интернет на скорости до 2 Мбит/с. Эта услуга уже успешно работает в Эссене и Мюльхайме, началось ее внедрение в Кельне и Дюрене. Довольно быстро ей удалось набрать несколько десятков тысяч клиентов, однако запланированный рубеж в 150 тыс. абонентов пока не преодолен. В Европе новая технология внедряется в Австрии и Шотландии. Австрийская энергетическая компания Tiwag подключила к своей сети около 1 тыс. клиентов.

Скорость передачи составляет 2 Мбит/с, но авторы проекта рассчитывают довести ее до 4,5 Мбит/с. В планах на будущее – включение в комплекс услуг и телефонной связи. В некоторых странах с неразвитой телекоммуникационной

структурой (например, Испании) создание сетей PowerLine является насущной необходимостью. По инициативе энергетической компании Scottish-Hydro-Electrics тестирование службы широкополосного доступа в Интернет через электросеть было запущено и в Шотландии. В настоящее время испытания ведутся в сельской местности. Как сообщает британское издание PC Advisor, в тестировании «Интернета через розетку» задействовано примерно 150 пользователей. Каждый абонент за 25 фунтов в месяц получил доступ в Интернет на скорости 2 Мбит/с, что гораздо выгоднее предложения, к примеру, интернет-провайдера Bulldog Communications (80 фунтов и 3 Мбит/с соответственно). Аналогичные проекты запущены в Италии и Швеции. В 2002 г. и московские энергетики провели ряд успешных испытаний новой технологии в Зеленограде. В марте 2003 г. на базе 19-го района «Московских кабельных сетей» была создана технологическая сеть связи, объединившая районный диспетчерский пункт с тремя питающими подстанциями, тремя распределительными подстанциями и одной трансформаторной. Результаты тестов показали, что эта технология доступа в Интернет вполне пригодна для создания локальных сетей передачи данных. С ее помощью удастся организовать доступ вентский канал успешно работает на расстоянии до 350 м от трансформаторной подстанции. Однако о фактах коммерческой эксплуатации технологии PowerLine для предоставления доступа в Интернет в России пока сведений нет. Есть лишь конкретные предложения для создания локальных компьютерных сетей в офисах с использованием имеющейся электропроводки. При помощи специальных адаптеров производства компании Ascom PowerLine организуется сеть на расстояниях до 200 м для 10–20 компьютеров. Не отстают и китайские разработчики. В апреле 2003 г. компания Fibrlink Networks, дочернее предприятие китайской государственной телекоммуникационной компании Power Telecom, приступила к экспериментальному предоставлению услуг доступа в Интернет через электрические сети в двух жилых зданиях Пекина. Абонентская плата составляет 12 долл. в месяц, без ограничения времени и трафика.

В Америке новая технология развивается не слишком активно (прежде всего, из-за уже имеющейся инфраструктуры связи), и все же в марте 2004 г. компания Cinergy образовала два совместных предприятия с компанией Current Communications Group для предоставления услуг доступа в Интернет через электросеть. Жители

штатов Огайо, Кентукки и Индиана получают широкополосный интернет-доступ и VoIP-телефонию по силовым проводам. Еще почти четыре десятка BPL-сетей доступа находятся в тестовой эксплуатации. В Америке новая технология, получившая название BPL (Broadband over Power Line), уже прошла испытания в Цинциннати (штат Огайо), где возможность нетрадиционного подключения получили более сотни физических и несколько юридических лиц. Тестирование показало высокое качество предоставления услуг и большой коммерческий потенциал технологии. Пользователи BPL отметили высокую скорость передачи данных – свыше 3 Мбит/с, при этом месячная абонентская плата оказалась ниже, чем за существующие широкополосные решения. Об активности российских регулирующих органов в области сертификации PLC и BPL пока ничего не слышно, а вот в США федеральной комиссией по электросвязи (Federal Communications Commission, FCC) такие спецификации уже приняты в октябре 2004 г. Основная проблема, вставшая перед FCC, – разработка требований, учитывающих, с одной стороны, пожелания потенциальных BPL-провайдеров, с другой – претензии от

сотовых операторов и радиолюбителей на возможные помехи. Несмотря на это, привлекательность BPL-сетей возрастает ввиду заранее оговоренной организации таких востребованных сервисов, как помощь инвалидам или служба E911. На повестке дня остались вопросы урегулирования отношений между поставщиками электричества и коммуникационными компаниями. В IEEE (Институт разработчиков в области электроники и электричества) разрабатывается стандарт для BPL – IEEE 1675, принятие которого ожидается не ранее 2009 г.

Производители бытовой электроники тоже не остались в стороне. В начале января 2005 г. компании Mitsubishi Electric и Sony объявили о своем намерении создать альянс с целью внедрения технологии PLC в жизнь. По убеждению его участников, каждая квартира уже оборудована всем необходимым для организации локальной сети, пропускная способность которой позволяет передавать любые данные, вплоть до потока видеопотока или IP-телефонии. Дело лишь за созданием окончательного оборудования.

Вычислительные системы с программируемой структурой

Хорошевский В. Г., *чл.-корр. РАН*

Сибирское отделение РАН, khor@isp.nsc.ru

Исследования по созданию вычислительных средств, базирующиеся на модели коллектива вычислителей, были начаты в Институте математики (ИМ) Сибирского отделения АН СССР в начале 1960-х годов по инициативе математика и механика академика С. Л. Соболева. Непосредственным руководителем работ стал специалист в области вычислительной техники доктор технических наук Э. В. Евреинов. Успех в исследованиях ИМ СО АН СССР был бы немислим, если бы они не опирались на достижения советских и западных научно-технических школ и прежде всего на школы основоположника отечественной вычислительной техники академика С. А. Лебедева. Особенно плодотворное влияние и поддержку в становлении и развитии направления «Вычислительные системы с программируемой структурой» оказали академики В. А. Мельников, В. С. Бурцев, В. П. Иванников; члены-корреспонденты Л. Н. Королев, Ю. И. Митропольский, Г. Г. Рябов и другие ученики и последователи С. А. Лебедева.

Первая работа сотрудников ИМ СО АН СССР [1] о возможности построения вычислительных систем высокой производительности опередила американские публикации в данной области примерно на 6 месяцев. В середине 1960-х годов выходит в свет монография [2], обобщающая первые результаты работ ИМ СО АН СССР по функциональным структурам вычислительных систем и параллельному программированию; под руководством Э. В. Евреинова создаются первые ВС [3]: «Минск-222» (1965 – 1966 гг.) и управляющая вычислительная система для автоматизации научных исследований (1964 – 1967 гг.). К началу 1970-х годов завершается формирование концепции вычислительных систем (ВС) с программируемой структурой, как средств обработки информации, основанных на модели коллектива вычислителей. Уместно заметить, что первоначальное название рассматриваемых средств – «Однородные вычислительные системы» [2], в конце 1970-х годов заменяется названием

«ВС с программируемой структурой», так как оно точнее отражает архитектурные возможности систем – коллективов вычислителей. «Однородные ВС» и «ВС с программируемой структурой» следует рассматривать как синонимические термины.

Начиная с 70-х годов 20 столетия теоретические и проектные работы в Сибирском отделении АН СССР (ныне СО РАН) по вычислительным системам с программируемой структурой ведутся под руководством одного из разработчиков первой ВС с программируемой структурой «Минск-222» члена-корреспондента В. Г. Хорошевского. Главными направлениями работ становятся:

- архитектура вычислительных систем;
- теория структур ВС: анализ и синтез структур сетей межмашинных связей;
- теория функционирования ВС: организация оптимального (субоптимального, стохастически оптимального) функционирования ВС в моно- и мультипрограммных режимах (обслуживание потоков параллельных задач);
- надежность и живучесть (потенциальная и структурная) ВС;
- самодиагностика и самоконтроль ВС;
- отказоустойчивые параллельные вычислительные технологии;
- проектирование вычислительных систем;
- распределенные операционные системы;
- системы параллельного программирования;
- параллельные алгоритмы и программы для решения прикладных задач.

Работы по вычислительным системам из академической сферы распространяются в промышленность, под руководством автора данной статьи создается ряд систем: МИНИМАКС (1975 г.), СУММА (1976 г.), МИКРОС-1 (1986 г.), МИКРОС-2 (1992 г.), МИКРОС-Т (1996 г.). Выходят в свет публикации сотрудников СО АН СССР (СО РАН) и других организаций. В 1978 г. издательством «Наука» публикуется монография [3]. Академик С. Л. Соболев дал о ней следующий отзыв:

«Книга – фундаментальный труд по теории и практике высокопроизводительных систем, основанных на новых принципах обработки информации. Созданное и развитое авторами научное направление однородных вычислительных систем является стержнем книги. Концепция однородных вычислительных систем позволяет в максимальной степени исчерпать современные достижения технологии микропроцессоров.

Книга представляет большой интерес для широкого круга ученых и специалистов промышленности».

Значимость разделов этой книги, посвященных теории функционирования вычислительных систем, была оценена с достаточной полнотой только через 20–25 лет после её опубликования. Это объясняется тем, что только в конце 20 столетия появились реальные условия для создания вычислительных систем действительно с массовым параллелизмом.

Понятие о вычислительных системах с программируемой структурой

Вычислительные системы (ВС) с программируемой структурой – это распределенные средства обработки информации, основанные на модели коллектива вычислителей. Тип архитектуры ВС – MIMD; в системах заложена возможность программной перенастройки архитектуры MIMD в архитектуры MISD или SIMD.

Основная функционально-структурная единица вычислительных ресурсов в системах рассматриваемого класса – это *элементарная машина* (ЭМ). В качестве ЭМ могут быть использованы ЭВМ, вычислительные ядра, многоядерные микропроцессоры, вычислительные узлы (композиции микропроцессоров), оснащенные средствами межмашинной коммутации. Допускается конфигурирование ВС с произвольным числом ЭМ. Следовательно, ВС с программируемой структурой относятся к масштабируемым средствам обработки информации и допускают формирование конфигураций с массовым параллелизмом (Scalable Massively Parallel Architecture Computer Systems).

В системах с программируемой структурой диалектически сочетаются архитектурные свойства универсальных и специализированных средств обработки информации. Рассматриваемые ВС – это универсальные параллельные компьютеры, которые способны программно настраиваться под структуру и параметры решаемых задач. Отмеченные архитектурные свойства

ВС с программируемой структурой оценены математиком и механиком академиком Н. Н. Яненко: *«Чем шире класс задач, охватываемой специализированной машиной, тем сложнее её структура и как наиболее совершенную форму ЭВМ следует рассматривать ЭВМ с перестраиваемой архитектурой»* [4].

При построении ВС с программируемой структурой доминирующими являются следующие три принципа:

1. массовый параллелизм (параллельность выполнения большого числа операций);
2. программируемость (автоматическая перестраиваемость или реконфигурируемость) структуры;
3. конструктивная однородность.

Следует подчеркнуть, что принцип программируемости структуры ВС является таким же важным, какими в свое время были предложения С. А. Лебедева и Дж. фон Неймана относительно организации в ЭВМ автоматической модификации программ. *Принцип программируемости структуры требует, чтобы в ВС была реализована возможность “хранения” программного описания функциональной структуры и программной ее модификации (перенастройки) с целью достижения адекватности структурам и параметрам решаемых задач.*

Выделяют подкласс пространственно-распределенных ВС. В него включаются макросистемы – системы сложной конфигурации, в которых в качестве функциональных элементов выступают пространственно-рассредоточенные вычислительные средства, основанные на моделях вычислителя и коллектива вычислителей, и сети связи, обеспечивающие взаимный теледоступ между средствами обработки информации.

Пространственно-распределенная ВС – это объединение географически удаленных друг от друга сосредоточенных ВС, основанное на принципах:

1. *параллельности* функционирования ВС (способности нескольких или всех систем совместно и одновременно решать одну сложную задачу, представленную параллельной программой);
2. превалирующего использования массовых аппаратурно-программных средств и существующих компьютерных сетей, включая Internet;
3. совместимости (информационной и программной) ВС.

Пространственно-распределенные ВС в общем случае предназначаются для выполнения параллельных программ решения задач произвольной сложности (с произвольным объемом вычислений) в *монопрограммном* и *мультипрограммных режимах* (на рассредоточенных в пространстве ресурсах). Они должны быть приспособленными и для выполнения функций, присутствующих вычислительным сетям.

Первая распределенная ВС с программируемой структурой АСТРА [3] была создана ИМ СО АН СССР и Новосибирским электротехническим институтом. Работы по проектированию ВС были начаты в 1970 г., а первая модель была сдана в эксплуатацию в 1972 г. Было построено семейство моделей АСТРА, среди которых были городские и междугородные конфигурации (Новосибирск, Москва). Модели формировались из ЭВМ «Минск-32» и использовали телефонные каналы связи. Были выполнены проекты распределенных ВС и на базе машин третьего поколения семейства ЕС ЭВМ [3].

Целесообразно подчеркнуть, что современное информационное пространство – это локальные и распределенные корпоративные вычислительные сети и глобальная сеть Internet. Дальнейшим шагом в развитии архитектуры сетей должны были стать распределенные вычислительные системы, способные реализовать параллельные алгоритмы решения сложной задачи на географически рассредоточенных ЭВМ и ВС. В самом деле, в конце прошлого столетия получают распространение распределенные информационные технологии: метакомпьютинг (metacomputing), масштабируемый (scalable), глобальный (global), Интернет (Internet) компьютеринг и P2P-компьютинг (peer-to-peer computing).

В 1997 году Национальный научный фонд США инициировал программу развития информационных технологий PACI (Partnerships for Advanced Computational Infrastructure). В ходе работ по программе PACI возникла концепция GRID (Global Resource Information Distribution), изначально рассматриваемая как аналогия распределенной электротехнической инфраструктуры. Аксиоматически ясно, что GRID-системы должны предоставлять распределенным пользователям разнообразные услуги по обработке и хранению данных, и главное они должны быть способны реализовать параллельные алгоритмы решения суперсложных задач на своих рассредоточенных ресурсах.

Следует особо подчеркнуть, что для любого этапа развития индустрии обработки информа-

ции характерно создание суперкомпьютеров. К последним относятся вычислительные средства, обладающие рекордной эффективностью (производительностью, надежностью, живучестью и технико-экономической эффективностью) для фиксированного этапа. Суперкомпьютеры могут быть основаны на любой из архитектурных парадигм, однако в них всегда присутствует параллелизм. Архитектура современных высокопроизводительных ВС, как правило, отличается от своих изначальных канонов. Архитектура одних и тех же систем в зависимости от уровня рассмотрения их функциональных структур может выглядеть и как MISD, и как SIMD, и как MIMD. Таким образом, можно констатировать, что *мультиархитектура стала парадигмой при конструировании высокопроизводительных ВС.*

Обобщая опыт развития индустрии обработки информации, можно заключить, что независимо от изначальной архитектурной парадигмы фирмы-создатели суперкомпьютеров к началу 21 столетия перешли на платформу ВС с программируемой структурой.

В следующем разделе будут описаны функциональные структуры советских и российских систем с программируемой структурой.

Реализация вычислительных систем с программируемой структурой

Интерес к практической реализации ВС с программируемой структурой постоянно проявлялся, начиная с 60-х годов двадцатого века. Первоначально он поддерживался прежде всего необходимостью проверки теоретических основ построения ВС, необходимостью отработки архитектурных решений и функциональной структуры ВС, а также параллельных вычислительных технологий. Позднее возрастающую роль стал играть утилитарный компонент целей создания ВС, в 1970-х годах этот компонент стал превалировать над исследовательским. Последнее обосновывается потребностью в ВС, обладающих высокой производительностью, надежностью и живучестью.

Работы по построению ВС, основанных на принципах модели коллектива вычислителей, были инициированы в ИМ СО АН СССР в 1964 г.; вскоре в институте было организовано и мини-производство ВС.

Вычислительная система «Минск-222»

Система «Минск-222» [3, 5] – первая в мире ВС с программируемой структурой. В проекте «Минск-222» были отработаны архитектурные, технические и программные решения, значительная часть из которых была «канонизирована» разработчиками «не-фон-неймановских» вычислительных средств.

Система «Минск-222» была разработана и построена Отделением вычислительной техники ИМ СО АН СССР совместно с Конструкторским бюро завода им. Г. К. Орджоникидзе Министерства радиопромышленности СССР (г. Минск). Руководитель работ по созданию ВС «Минск-222» — Э. В. Евреинов; основные разработчики: В. Г. Хорошевский, Б. А. Сидристый, Г. П. Лопато, А. Н. Василевский. Работы по проектированию ВС «Минск-222» были начаты в 1965 г., а первый ее образец был установлен в апреле 1966 г. в Институте математики АН БССР. Системы «Минск-222» были смонтированы в нескольких организациях Советского Союза и эксплуатировались более 15 лет.

Архитектура ВС:

- МIMD-архитектура, распределенность ресурсов;
- параллелизм, однородность, программируемость структуры;
- одномерная (кольцевая) топология;
- масштабируемость: 1–16 элементарных машин (ЭМ);
- использование промышленных ЭВМ второго поколения.

Элементарная машина ВС

В системе «Минск-222» каждая ЭМ состояла из вычислительного модуля (ВМ) и системного устройства (СУ). В качестве ЭМ были использованы серийные ЭВМ «Минск-2» или «Минск-22», выпускавшиеся заводом им. Г. К. Орджоникидзе (г. Минск). Указанные ЭВМ имели одну и ту же двухадресную архитектуру, «Минск-22» в сравнении с «Минск-2» обладала магнитной оперативной памятью удвоенной емкости (8 К 37-разрядных слов).

Подход к построению параллельных ВС, ориентированный на применение серийных ЭВМ, был впервые применен в Сибирском отделении АН СССР [3], а не за рубежом (разработки 1970-х годов Университета Карнеги—Меллона, современные кластерные ВС).

В состав *системного устройства* (рис. 1) входили локальный коммутатор (ЛК) каналов связи и блок операций системы (БОС). Коммутатор ЛК_i

состоял из вентилях, которые открывали или закрывали канал связи, идущий к соседней справа ЭМ, т. е. к коммутатору ЛК_j, где $j = (i + 1) \bmod N$. Вентили управлялись сигналами, поступавшими из БОС.

Блок операций системы включал в себя регистр настройки (РН) и узел, реализующий системные команды. Содержимое РН определяло вид соединительной функции коммутатора и степень участия ЭМ при системных взаимодействиях. Регистр настройки состоял из трех разрядов: $TR, TQ, T\Omega$

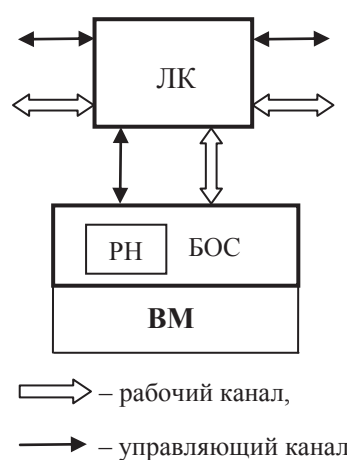


Рис. 1. ЭМ системы «Минск-222».

Триггер TR позволял разбивать систему на функционально изолированные подсистемы. Триггеры TQ и $T\Omega$ конкретизировали степень участия машин в выполнении некоторых системных команд.

Системное устройство было реализовано на 80 стандартных элементах и составляло менее 1,5 % объема оборудования АЛУ и устройства управления ЭВМ «Минск-22».

Системные команды ВС

К системным относятся команды, обеспечивающие организацию и реализацию параллельных вычислительных процессов, в частности, обменов управляющей информацией и данными между ветвями параллельной программы. Набор системных команд ВС «Минск-222» составляли *команды настройки, обмена, обобщенных безусловного и условного переходов* [3, 5]. *Команды настройки* позволяли программировать структуру ВС и задавать степень участия каждой ЭМ в реализации параллельных процессов.

Команды обмена — это команды передачи (П) и приема (ПР). По команде П передающая ЭМ выдавала в (настроенный) канал межмашин-

ной связи определенное количество слов из своей памяти. Такой способ организации обменов информацией между машинами ВС подобен функционированию радио- и телепередающих систем. Он не зависит от числа ЭМ в системе и позволяет избежать трудностей с адресацией машин.

Команды обобщенного безусловного перехода (ОБП) предназначались для принудительного управления работой ВС. Управляющая машина при выполнении ОБП засылала в канал содержимое заданной ячейки своей памяти, которое воспринималось как команда только ЭМ, содержащими единицу в разрядах TQ регистров настройки. Команды ОБП позволяли осуществлять иницирование работы ВС и ее загрузку данными из любой ЭМ, а также вмешиваться в параллельные вычислительные процессы и принудительно управлять работой подмножеств машин системы из любой ЭМ. Следовательно, при помощи команды ОБП из любой ЭМ можно было иницировать программирование структуры ВС.

Команды обобщенного условного перехода (ОУП) и содержимое TQ регистров настройки элементарных машин позволяли управлять параллельными вычислительными процессами. Одна из команд ОУП служила для синхронизации параллельных процессов (машин, выполняющих ветви параллельной программы). По сути эта команда позволяла реализовать механизм синхронизации, получивший позднее название "Bartier". Команды ОУП использовались, в частности, для организации ветвлений в параллельных вычислениях, для реализации параллельных циклов.

Программное обеспечение «Минск-222»

В системе «Минск-22» ПО было ориентировано лишь на реализацию в монопрограммном режиме параллельных программ (*P-программ*) решения сложных задач. Оно состояло из двух частей: *системы P-программирования* и *пакета прикладных адаптирующихся P-программ*. Система параллельного *P-программирования* включала средства автоматизации *P-программирования*, отладки, редактирования и анализа *P-программ*.

Средства автоматизации P-программирования — языки и трансляторы. В качестве входных в системе «Минск-222» использовались расширенные языки [5]: автокод АКИ, ЛЯПАС, ALGOL, BASIC. В расширенные языки были включены средства для описания взаимодействий между параллельными ветвями вычислений. Каждый транслятор для ВС состоял из двух частей: обычного транслятора для ЭВМ и

системного блока для реализации межмашинных взаимодействий в ВС. Любой системный блок представлял собой совокупность программ для реализации операций настройки, обмена, ОБП и ОУП, которые были включены в библиотеки трансляторов.

Средства отладки и редактирования P-программ — совокупность четырех стандартных программ. Первая программа преобразовывала отлаживаемую *P-программу* в последовательную и выявляла ошибки, не связанные с использованием системных команд. Вторая программа служила для моделирования на одной машине выполнения *P-программы* из двух ветвей. Всевозможные (допустимые и недопустимые) взаимодействия ветвей были представлены матрицей переходов к моделирующим или авостным подпрограммам. Эта же программа могла определять время простоев машин, время работы отдельных блоков *P-программы* и точность вычислений. Третья программа позволяла вывести на печать заданное количество раз содержимое интересующих областей памяти перед выполнением команд обмена в процессе контрольной реализации параллельной программы на ВС «Минск-222». Четвертая программа служила для корректировки *P-программ*.

Средства анализа P-программ были представлены тремя программами. Первая программа служила для анализа распределения памяти между блоками исследуемой программы. Вторая программа предназначалась для измерения времени простоев машин ВС. Третья программа применялась для измерения времени работы участков *P-программы*.

Пакеты прикладных адаптирующихся P-программ были ориентированы на решение задач повышенной сложности. Параметры таких задач, как правило, не позволяли решать их на ЭВМ «Минск-22» за удовлетворительное время.

Из опыта создания ПО для системы «Минск-222» установлено, что его объем отличается от объема программного обеспечения ЭВМ «Минск-22» не более чем на 10 %.

Опыт эксплуатации ВС «Минск-222»

Полный набор схем обмена информацией между ветвями *P-программ*, как показала эксплуатация ВС «Минск-222», составляют: дифференцированный (ДО), трансляционный (ТО), трансляционно-циклический (ТЦО), конвейерно-параллельный (КПО) и коллекторный (КО) обмены.

При трансляционном обмене (*One-to-all Broadcast*) осуществляется передача одной и той же информации из одной (любой) ветви одновременно во все остальные ветви P -программы. Трансляционно-циклический обмен (*All-to-all Broadcast*) реализует трансляцию информации из каждой ветви во все остальные. Следовательно, если трансляционный обмен выполняется за 1 такт, то трансляционно-циклический — за n тактов. Конвейерно-параллельный обмен обеспечивает передачу информации между соседними ветвями; он выполняется за два такта. Так, например, при четном n в первом такте осуществляется передача информации из ветвей $P_1, P_3, \dots, P_{i-1}, \dots, P_{n-3}, P_{n-1}$ соответственно в ветви $P_2, P_4, \dots, P_i, \dots, P_{n-2}, P_n$; во втором такте информация из последней последовательности ветвей поступает соответственно в ветви $P_3, P_5, \dots, P_{i+1}, \dots, P_{n-1}, P_n$. Коллекторный обмен представляет собой инвертированный трансляционный обмен, в одну ветвь последовательно собирается информация из $l < n$ ветвей. Такой обмен требует l тактов и реализуется как последовательность из l дифференцированных обменов.

Было установлено, что при решении задач на ВС «Минск-222» системные команды в P -программах составляли, как правило, менее 10% их общего объема. Следовательно, можно считать, что затраты при разработке параллельных и эквивалентных им последовательных программ имеет один и тот же порядок. Выявлено также, что для ВС «Минск-222» доля затрат времени на системные взаимодействия (включая синхронизацию) составляет, как правило, несколько процентов, что является следствием применения мето-

дики крупноблочного распараллеливания задач. Кроме того, выяснилось, что за счет большей емкости оперативной памяти в системе «Минск-222» по сравнению с одной ЭВМ «Минск-22» и за счет быстрого действия каналов связи, сравнимого с быстрым действием ЭВМ при выполнении операций, получается дополнительный значительный выигрыш во времени решения задач на ВС.

Следует особо подчеркнуть, что архитектурные решения, реализованные в ВС «Минск-222», стали, по сути, каноническими. Схемы обмена информацией между ветвями P -программ и рассмотренные системные команды нашли отражение в современной инструментрии, используемой при построении распределенных и параллельных ВС. Так, в MPI (Message Passing Interface) — библиотеке функций, предназначенной для поддержки параллельных процессов — применяются как дифференцированный (Point-to-point Communication), так и коллективные взаимодействия (Collective Communications). В табл. 1 приведены основные виды системных взаимодействий и реализующие их команды ВС «Минск-222» и функции MPI.

Полученный опыт по проектированию, математической и технической эксплуатации «Минск-222» был использован в последующих проектах вычислительных систем с программируемой структурой.

Вычислительные системы из мини-ЭВМ

Вычислительные системы, которые формировались из аппаратурно-программных средств мини-ЭВМ, относились к группе мини-ВС. Построение таких ВС было одной из основных ми-

Таблица 1.

Вид взаимодействия	Команды ВС «Минск-222»	Функции MPI
Дифференцированный обмен	П, ПР	MPI_Send, MPI_Recv
Трансляционный обмен	П, ПР	MPI_Bcast
Трансляционно-циклический обмен	П, ПР, n итераций	MPI_Alltoall или MPI_Allscatter
Коллекторный обмен	П, ПР, $n - 1$ итерация	MPI_Gather
Синхронизация элементарных машин	ОУП	MPI_Barrier
Разбиение ВС на подсистемы	Н	MPI_Comm_group MPI_Group_incl MPI_Comm_create MPI_Cart_create

ровых тенденций развития вычислительной техники 1970-х годов. Опыт эксплуатации показал, что при решении большого круга задач мини-ВС были более эффективны с точки зрения производительности, надежности, живучести и стоимости, чем одна или даже несколько больших ЭВМ.

Работы по созданию ВС из мини-машин достаточно интенсивно велись в США. Однако общей концепции построения таких систем американские специалисты не выработали. Анализ проектов показывает, что использовались в основном три способа организации ВС: системы с общей памятью; ВС с общей шиной (или системой шин), к которой подключались процессоры, запоминающие и другие устройства; системы, в которых машины взаимодействовали через общую группу устройств ввода-вывода информации. Как правило, системы не имели программируемой структуры и обладали ограниченными возможностями к наращиванию.

При создании мини-ВС в Советском Союзе за основу была взята концепция ВС с программируемой структурой. Архитектурные решения в области мини-ВС, опыт их проектирования, разработки системного и прикладного ПО нашли массовое применение только в конце двадцатого века. Именно вычислительные кластеры являются по существу многопроцессорными или многомашинными ВС, конфигурируемыми из микропроцессоров или персональных ЭВМ (например, IBM PC). При этом заметна архитектурная близость мини-ЭВМ и современных персональных компьютеров.

Вычислительная система МИНИМАКС

МИНИМАКС (МИНИМАшинная программно Коммутируемая Система) создана ИМ СО АН СССР (Отделом вычислительных систем) и Научно-производственным объединением «Импульс» Министерства приборостроения, средств автоматизации и систем управления СССР (г. Северодонецк). Технический проект МИНИМАКС разработан в 1974 г., а опытно-промышленный образец системы был изготовлен и отработан в 1975 г.

Архитектура системы:

- МИМД-архитектура;
- распределенность средств управления, обработки и памяти;
- параллелизм, однородность, модульность;
- программируемость структуры;
- двумерная (циркулянтная) топология;

- масштабируемость;
- живучесть;
- максимальное использование промышленных средств мини-ЭВМ.

Функциональная структура мини-ВС МИНИМАКС — композиция из произвольного количества элементарных машин и программно настраиваемой сети связей между ними. Взаимодействия между ЭМ в системе МИНИМАКС осуществлялись через сеть связей (рис. 2), которая формировалась из одномерных 1 и двумерных 2 полудуплексных каналов. Одномерные каналы связи 1 были управляющими; они служили для программирования соединений между ЭМ по каналам связи 2, а также для передачи между ЭМ управляющей информации, регламентирующей использование общих ресурсов (внешних устройств, сервисных программ, файлов и т. п.). Двумерные каналы связи 2 являлись рабочими; они применялись для следующих целей: реализации основных межмашинных взаимодействий, пересылки массивов данных между памятьми передающей ЭМ и одной или нескольких принимающих ЭМ, передачи адресов из одной ЭМ в другую и обмена логическими переменными между машинами.

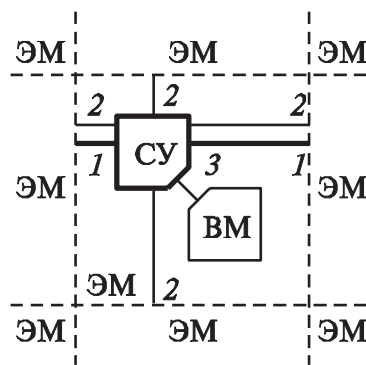
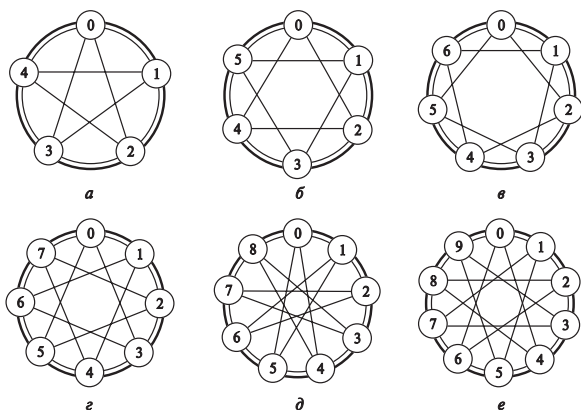


Рис. 2. ЭМ мини-ВС МИНИМАКС.

Межмашинные взаимодействия при функционировании мини-ВС реализовывались с помощью специальных подпрограмм — *системных драйверов*, которые, в свою очередь, использовали специальные команды (занесение кода на регистр настройки, считывание его содержимого, занесение информации в системное устройство (СУ0 о начальном адресе передаваемого массива данных и т. п.).

Очевидно, что допускалось формирование в МИНИМАКС тороидальных структур; для специальных областей применения использовались оптимальные графы:



В пределах мини-ВС МИНИМАКС допускалось формирование произвольного числа подсистем из любого количества ЭМ. Подсистему составляли взаимодействовавшие друг с другом ЭМ вместе с машинами, которые использовались в качестве транзитных пунктов передачи информации.

Элементарная машина МИНИМАКС – это композиция из ВМ и СУ. Структура ЭМ данной мини-ВС не была жестко заданной и определялась областью применения. Состав каждой ЭМ допускал варьирование; компоновка ЭМ проводилась по правилам, которые были приняты для агрегатных средств ВТ на микроэлектронной основе (АСВТ-М) или для средств системы малых ЭВМ (СМ ЭВМ). В качестве ВМ могли быть использованы любые кон-фигурации мини-ЭВМ на базе процес-соров М-6000, М-7000, СМ-1П. Архитектура системы МИНИМАКС была рассчитана также на применение мини-ЭВМ моделей HP 2114—2116 семейства Hewlett—Packard.

Системное устройство было спроектировано как автономное устройство АСВТ-М. Оно подключалось к ВМ через связи 3 (рис. 5). При выборе способа реализации связей 3 учитывались принципы построения АСВТ-М и следующие отсюда ограничения:

- целесообразность построения системного устройства в виде отдельного модуля;
- недопустимость изменений в схемах и конструкции процессоров АСВТ-М.

Программное обеспечение мини-ВС МИНИМАКС состояла из управляющей системы, средств *P*-программирования, пакетов

P-программ и комплекса программ технического обслуживания.

Вычислительная система СУММА

В 1970-х годах для управления процессами в реальном времени широко применялись не только мини-машины, но и вычислительные сети и системы из мини-ЭВМ. В данном разделе описывается вторая отечественная мини-ВС: Система Управляющая Мини-Машинная (СУММА).

Система СУММА была разработана ИМ СО АН СССР (Отделом вычислительных систем) совместно с Производственным объединением «Кварц» Министерства электронной промышленности СССР (г. Калининград). В ПО «Кварц» позднее производилась суперЭВМ «Электроника СС БИС-1», разработанная под руководством академика В. А. Мельникова. Так в ПО «Кварц» произошло слияние интересов двух научных школ по достижению их главной цели – создать в СССР индустрию суперкомпьютеров.

Техническое проектирование мини-ВС было выполнено в 1975 г., опытно-промышленный образец был изготовлен и отработан в 1976 г.

Данная Мини-ВС, как и система МИНИМАКС, имела программируемую структуру и свои архитектурные особенности:

- единый канал для управляющей и рабочей информации;
- аппаратно-программную реализацию системных взаимодействий.

Функциональная структура мини-ВС СУММА характеризовалась большой гибкостью. Ее можно было легко расширить или сократить в соответствии с предъявляемыми требованиями. Принципиальные ограничения на структуру мини-ВС (количество ЭМ и порядок их соединения) не накладывались, однако при любой структуре каждая ЭМ могла взаимодействовать не более чем с тремя соседними машинами с помощью полудуплексных каналов (рис. 3). В мини-ВС была заложена возможность «программировать» адресацию ЭМ, в частности, система могла быть настроена на относительную адресацию ЭМ.

Системы управления, на применение в которых была рассчитана мини-ВС СУММА, характеризуются стабильностью решаемых задач, жесткими требованиями к реактивности на изменение операционной обстановки (преимущественно детерминированный поток запросов на обслуживание). Следовательно, в системах управления перепрограммирование структуры мини-ВС требова-

лось выполнять редко, и время обмена управляющей информацией в общем времени работы машин системы составляло незначительную часть. Эти факторы позволили ограничиться единым каналом для обмена управляющей (настроечной) информацией и данными между ЭМ мини-ВС.

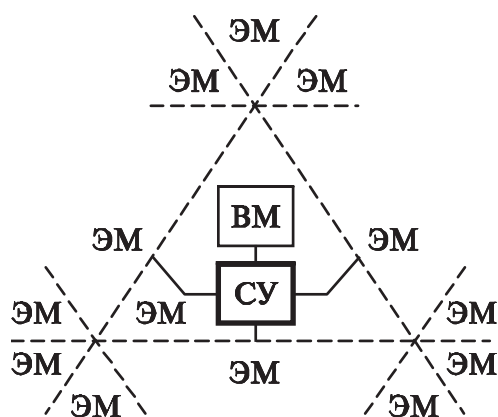


Рис. 3. ЭМ мини-ВС СУММА.

К системам управления предъявляются повышенные требования по живучести, следовательно, их вычислительные средства должны обладать структурной живучестью. Для формирования мини-ВС СУММА использовались оптимальные графы (рис. 4).

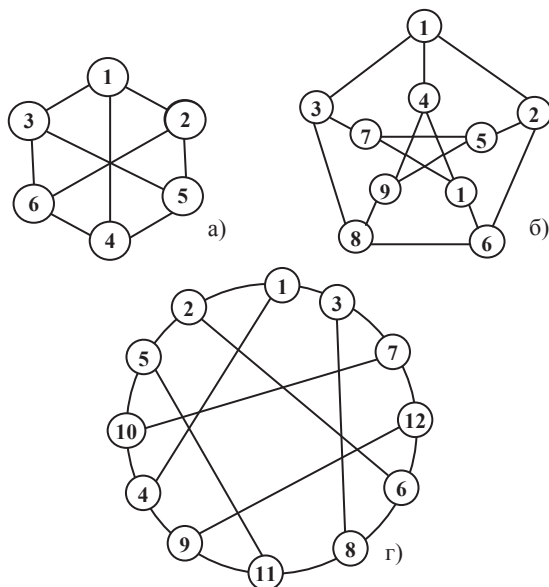


Рис. 8. Оптимальные структуры мини-ВС СУММА.

Элементарная машина системы СУММА формировалась как «трехполюсник», или точнее, композиция из ВМ и СУ, рассчитанного на три межмашинные связи (рис. 3).

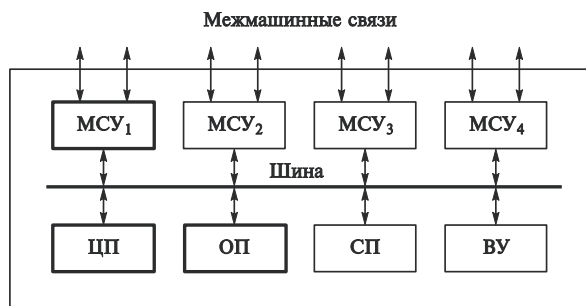
Вычислительный модуль предназначался для выполнения всех операций, связанных с переработкой информации, в частности для иницирования реализации системных операций. Системное устройство использовалось для реализации системных взаимодействий машин, в частности для программирования структуры мини-ВС. В качестве ВМ использовали произвольные конфигурации мини-ЭВМ «Электроника-100 И». Следует заметить, что архитектура системы СУММА была ориентирована также на применение мини-ЭВМ PDP-8 фирмы Digital Equipment Corp.

Программное обеспечение мини-ВС СУММА – проблемно-ориентированное. В его состав входили супервизор (являвшийся программой управления процессами в реальном масштабе времени), система P-программирования (включающая макроассемблер MACRO-8P), системы для автоматизированного управления технологическими процессами и комплекс программ технического обслуживания.

Вычислительные системы семейства МИКРОС

Прогресс в вычислительной технике неразрывно связан с достижениями в области элементной базы и в интегральной технологии. В конце 1970-х годов мини-процессоры вытесняются микропроцессорами, на смену мини-ЭВМ пришли микроЭВМ; в индустрии обработки информации создаются параллельные ВС как коллективы микропроцессоров

В начале 1980-х годов в Отделе вычислительных систем СО АН СССР иницируются работы по научно-исследовательскому проекту МИКРОС [6], целью которых было создание МИКРОпроцессорных Систем с программируемой структурой (МИКРОС). Результатом работ явилось семейство МИКРОС, включающее модели МИКРОС-1 (1986); МИКРОС-2 (1992); МИКРОС-Т (1996). Разработка моделей семейства МИКРОС осуществлялась в содружестве с подразделениями Научно-производственного объединения «Алмаз» и Научно-исследовательского института «Квант» Министерства радиопромышленности СССР (г. Москва).



Функциональная структура МИКРОС

Возможности систем семейства МИКРОС определяются количеством ЭМ, входящих в их состав, конфигурациями ЭМ и топологией сетей межмашинных связей. Количество ЭМ в любой из моделей (МИКРОС-1, МИКРОС-2, МИКРОС-Т) не фиксировано. Каждая ЭМ — это многополюсник, число полюсов ν в первых моделях систем составляло от 2 до 8, а в модели МИКРОС-Т $\nu = 4$.

Каждая генерация ВС семейства МИКРОС адекватно учитывала текущие возможности ВТ и интегральной технологии. Для формирования конфигураций ЭМ моделей МИКРОС-1 и МИКРОС-2 использовались средства микроЭВМ отечественного семейства «Электроника»; элементарная машина (рис. 5) представлялась композицией из модулей микроЭВМ и модулей системного устройства (МСУ).

Рис. 5. ЭМ систем МИКРОС-1 и МИКРОС-2.

Свойством масштабируемости обладали не только модели семейства МИКРОС, но и их ЭМ. Простейшая конфигурация ЭМ состоит из одного МСУ, центрального процессора (ЦП) и оперативной памяти (ОП). Модуль СУ обеспечивал реализацию системных операций в ВС и непосредственную связь данной ЭМ с двумя соседними машинами через полудуплексные каналы. Модуль СУ позволял использовать в качестве каналов различные средства, в частности, экранированные провода (при расстоянии между ЭМ до 30 м), либо радиочастотные кабели (если расстояние между ЭМ не превышало 300 м), либо коммутируемые или выделенные телефонные каналы связи (с использованием аппаратуры передачи данных независимо от расстояния между ЭМ). Заложенная в модуль СУ схема обеспечения связности машин была равно пригодна для формирования пространственно сосредоточенных и распределенных ВС.

В моделях ВС МИКРОС-1 и МИКРОС-2 в качестве базовых машин были использованы микроЭВМ «Электроника 60М» и «Электроника

60-1» соответственно. Расширенные конфигурации ЭМ систем МИКРОС-1 и МИКРОС-2 могли иметь до четырех модулей СУ, специальный процессор (СП), дополнительные модули оперативной памяти, набор внешних устройств (ВУ). *Специальные процессоры* «Электроника МТ-70» или «Электроника 1603» расширяли вычислительные возможности ЦП при решении научно-технических задач, связанных с обработкой значительных массивов данных и с выполнением больших объемов однородных вычислений.

Модули системного устройства для системы МИКРОС-2 обладали большими функциональными возможностями, чем в системе МИКРОС-1. Их аппаратура, в частности, позволяла осуществлять: обработку входных/выходных запросов для межмашинных связей; анализ семафоров; формирование пакетов выходных сообщений; управление входными и выходными портами при выполнении системных команд; мультиадресные передачи информации; совмещение межмашинных обменов информацией с вычислениями.

Система МИКРОС-Т базируется на транзьютерных технологиях [7]. Такие технологии позволяют формировать двумерные ВС с массовым параллелизмом. Двумерные структуры ВС формируются путем отождествления полюсов-линков (Link — связь).

Простейшая конфигурация ЭМ представляется транзьютером (например, Immos T 805) с памятью, развитые конфигурации ЭМ могли включать в себя: высокопроизводительные микропроцессоры — Intel 860 (компания Intel), PowerPC (альянс компаний IBM, Apple и Motorola), Alpha (компания DEC и Compaq) и др. Для формирования ЭМ системы МИКРОС-Т могли быть использованы стандартные решения зарубежных и отечественных фирм-производителей транзьютерных модулей.

Программное обеспечение МИКРОС

Эффективная работа ВС и ее пользователей немаловажна без операционной системы (ОС) и среды параллельного программирования. Любая система семейства МИКРОС, как и ее ПО, были открыты к совершенствованию. Ряд моделей семейства ВС (МИКРОС-1, МИКРОС-2 и МИКРОС-Т) породил и соответствующий ряд генераций ПО [17].

В основу *операционной системы* МИКРОС положены следующие принципы:

- независимость от числа машин в ВС;
- модульность построения;

- распределенность модулей по машинам ВС и децентрализованное их использование;
- локальность связей между модулями;
- асинхронность взаимодействий модулей;
- развиваемость (изменяемость и пополняемость состава модулей, в частности возможность замены программных модулей на аппаратные);
- иерархичность построения: разделение системы на уровни, каждый из которых строится на основе предыдущих и освобождает пользователя от специфических для уровня операций по погружению задачи в систему;
- преемственность с ОС базовых микропроцессорных средств (либо микроЭВМ «Электроника», либо транспьютеров, в зависимости от моделей семейства ВС МИКРОС).

Все созданные генерации ОС (МИКРОС-1, МИКРОС-2, МИКРОС-Т) являются распределенными и децентрализованными. Децентрализованная распределенная ОС МИКРОС способна функционировать в ВС произвольной конфигурации; ОС создает в каждой ЭМ «окружение», позволяющее осуществлять динамическую настройку адаптирующейся параллельной программы на существующую конфигурацию ВС (или подсистемы). Децентрализованные процедуры маршрутизации обеспечивают передачу сообщений между любыми ЭМ системы. Указанные свойства ОС МИКРОС являются основой для поддержки живучести ВС (и, следовательно, для организации отказоустойчивых вычислений).

В среде *программирования* МИКРОС имеются языки параллельного программирования *P-ФОРТРАН* и *P-C*. Эти языки построены путем расширения соответствующих традиционных языков FORTRAN и C примитивами организации межмашинных взаимодействий и примитивами оценки параметров подсистем, на которых исполняются параллельные программы. Первые позволяют организовать взаимодействия между любыми ветвями программы, вторые дают возможность использовать параметры подсистемы для адаптации программы к текущей конфигурации подсистемы. Это свойство существенно с двух точек зрения: простоты организации параллельных вычислений и отказоустойчивости. Реализация данных примитивов основывается на средствах распределенной децентрализованной операционной системы МИКРОС.

Таким образом, ВС семейства МИКРОС основываются на перспективных принципах обработки информации, строятся из аппаратно-

программных средств микропроцессорной техники, обладают гибкими возможностями по статической и динамической реконфигурации своих структур, позволяют достичь высокой производительности, надежности и живучести в широкой области применения.

Продолжением ряда ВС МИКРОС-1, МИКРОС-2 и МИКРОС-Т являются высокопроизводительные ВС с массовым параллелизмом семейства МВС.

Опыт, приобретенный при создании мини-ВС и микропроцессорных систем, может быть положен в основу будущих разработок суперВС как ансамблей микропроцессоров, размещаемых на крупномасштабных полупроводниковых пластинах.

Пространственно-распределенная мультикластерная ВС

Кластерные ВС — параллельные средства обработки информации, интуитивная оценка архитектурных возможностей которых вытекает из семантики слова кластер (Cluster — группа). Такие системы получили широкое распространение уже в 1990 годах. В списке Top500 кластерные системы доминируют; их количество приближается к 400.

Термин «вычислительный кластер», по видимому, был впервые введен DEC (Digital Equipment Corporation). По определению DEC, кластер — это группа компьютеров, которые связаны между собой и функционируют как единое средство обработки информации. Из приведенного определения видно, что корпорация DEC, по сути, ввела синоним термину «вычислительная система», а не особый тип средств обработки информации. Для создания кластерных ВС используются и MISD-, и SIMD-, и MIMD-архитектуры, различные функциональные структуры и конструктивные решения.

В наиболее общей трактовке *кластерная ВС, или кластер*, — это композиция множества вычислителей, сети связей между ними и программного обеспечения, предназначенная для параллельной обработки информации (в частности, реализации параллельных алгоритмов решения сложных задач). При формировании кластерной ВС могут быть использованы как стандартные промышленные компоненты, так и специально созданные средства. Однако в кластерных ВС, как правило, преобладают массовые аппаратно-программные средства. Последнее, по существу,

является принципом конструирования кластерных ВС, обеспечивающим их высокую технико-экономическую эффективность.

Начало 21 века ознаменовалось созданием сосредоточенных мультикластерных ВС (IBM RoadRunner состоит из 18 кластеров) и характеризуется переходом от «виртуальных метакомпьютеров» (использующих ресурсы нескольких суперкомпьютеров) к большемасштабным пространственно-распределенным мультикластерным ВС как макроколлективам рассредоточенных кластеров, взаимодействующих между собой через локальные и глобальные сети (включая всемирную сеть Internet).

Объединенным коллективом Лаборатории вычислительных систем Института физики полупроводников им. А. В. Ржанова СО РАН и Центром параллельных вычислительных технологий Сибирского государственного университета телекоммуникаций и информатики (СибГУТИ) создана масштабируемая GRID-модель – пространственно-распределенная мультикластерная ВС.

Текущая терафлопсная конфигурация пространственно-распределенной ВС имеет в своем составе более 10 кластеров, расположенных в институтах СО РАН и СибГУТИ. Для формирования кластеров использовались, в частности, процессоры AMD Opteron и Intel Xeon (включая двух- и четырехядерные). Система оснащена операционной системой GNU/Linux и специально разработанными подсистемами: мультипрограммирования, оптимизации вложения параллельных программ в ВС, анализа MPI-программ, самоконтроля и самодиагностики ВС, организации распределенной очереди задач, удаленного доступа и мониторинга ВС.

В вычислительной системе имеются средства разработки *последовательных программ*:

- компиляторы: GNU GCC, SUN, Intel;
- библиотеки: GNU GSL, AMD ACML, Intel MKL;
- средства отладки и анализа;

и параллельных программ:

- MPI: MPICH2, OpenMPI, Intel MPI;
- OpenMP: GCC, SUN, Intel;
- средства отладки и анализа: VampirTrace.

Пространственно-распределенная ВС используется не только в научных исследованиях, но и при подготовке специалистов в области GRID- и параллельных вычислительных технологий.

Заключение

Современные суперВС (пространственно сосредоточенные и распределенные) полностью основываются на модели коллектива вычислителей.

Концепция ВС с программируемой структурой позволяет создавать технико-экономически эффективные средства обработки информации, обладающие высокой производительностью, надежностью и живучестью.

Литература

1. Евреинов Э. В., Косарев Ю. Г. О возможности построения вычислительных систем высокой производительности. – Новосибирск: СО АН СССР, 1962. – 39 с.
2. Евреинов Э. В., Косарев Ю. Г. Однородные универсальные вычислительные системы высокой производительности. – Новосибирск: Наука, 1966. – 308 с.
3. Евреинов Э. В., Хорошевский В. Г. Однородные вычислительные системы. – Новосибирск: Наука, 1978. – 320 с.
4. Яненко Н. Н. Перспективы развития вычислительной математики на основе вычислительных систем // Препринт “ЭВМ. Перспективы и гипотезы”. – Новосибирск: ИТПМ СО АН СССР, 1981. № 46. С. 3–6.
5. Хорошевский В. Г. Архитектура вычислительных систем. – 2-е изд., перераб. и доп. – М.: МГТУ им. Н. Э. Баумана, 2008. – 520 с.
6. Хорошевский В. Г. Вычислительная система МИКРОС // Препринт. – Новосибирск: ИМ СО АН СССР, 1981. № 38 (ОВС-19). – 52 с.
7. Транспьютеры. Архитектура и программное обеспечение / Под ред. Г. Харпа. – М.: Радио и связь, 1993. – 303 с.

Влияние программирования на архитектуру вычислительных систем (от прошлого к будущему)

Королев Л. Н., *чл.-корр. РАН*

Зав. Кафедрой МГУ им. М. В. Ломоносова

В 2005 году исполнилось 60 лет с момента ввода в строй машины ENIAC, которую считают первой в мире ЭВМ с программным управлением. Все эти годы в условиях технических ограничений элементной базы шел процесс совершенствования архитектуры компьютеров в направлении максимального повышения производительности и повышения эффективности планирования обработки данных, то есть в направлении аппаратной поддержки программирования.

В преддверии наступления нового тысячелетия крупнейшие фирмы объявили о создании новых архитектурных линий, которые должны удовлетворить требованиям к вычислительной технике, выдвигаемым со стороны всех сфер научно-технического прогресса. Имеется в виду Alpha DEC, POWER IBM, SPARC SUN, HP-PA, IA-64 и Pentium Intel и др.

Несмотря на то, что в какой-то степени в новом столетии процессоры этих линий сохранили ядро архитектуры «внешней» системы команд, их микроархитектура заметно изменилась.

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. Архитектуры типа VLIW-EPIC доминируют в современных разработках.

При этом 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 мультиме-

дных данных, что увеличило число типов данных, поддерживаемых аппаратурой. Возникла существенная необходимость отражения в архитектуре технологии создания кластеров и машин с массовым параллелизмом на основе использования микропроцессоров. В результате появились процессоры, логическая сложность которых исчисляется в сотни миллионов логически активных элементов (CELL – 235 млн.).

Возникает вопрос, не приведет ли экстенсивный путь развития архитектур процессоров, основанный на лозунге: «логических элементов - не жалеть, аппаратные динамические интерпретаторы и системные программисты сделают всё что нужно» к тому, что уже было. А именно, RISC-архитектуры заменили собой CISC-архитектуры по той простой причине, что трансляторы использовали приблизительно только 20% из всего набора представленных в процессоре команд!

Несомненный интерес представляют 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: «выполни действие сразу в тот момент, когда будут готовы операнды», как представляется, позволит достичь максимальной эффективности динамического распараллеливания более простыми и логически прозрачными средствами, чем нынешние VLIW-EPIC архитектуры. Существенно, что в этих разработках, наконец, удалось отказаться от привычной парадигмы линейного характера физической оперативной памяти.

Несомненный интерес представляют работы, выполненные в свое время в ИТМ и ВТ по разработке архитектуры МКП (А. А. Соколов, А. Ю. Бяков и др.). Если следовать современной терминологии, это была двужадерная суперскалярная архитектура процессора, включавшая блок

параллельной манипуляции данных 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 процессор-процессор, и 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.

Экспериментальные исследования с использованием отечественной вычислительной техники и методов некорректно поставленных задач

Горячев В. А., к. ф. -м. н. , НИИСИ РАН

Уважаемые коллеги, цель данного сообщения состоит в том, чтобы кратко показать тот круг проблем вычислительной техники и программирования, свидетелем и участником которых с середины прошлого века мне посчастливилось оказаться. В частности, как выпускник кафедры ТЯФ МИФИ в 1969 г. , выполнявший УИР в ФИАНе, был распределен на работу в отраслевой НИИТ МСМ (мин. Атом), занимавшийся экспериментальными исследованиями. Тогда на «вооружении» этих исследований в известных КБ Арзамаса-16 (ВНИИЭФ) и Снежинска (ВНИИП) эксплуатировались отечественные супер-ЭВМ: М-20, БЭСМ-6 и др. , разработанные В. А. Мельниковым и руководимым им коллективом.

Суть решаемых проблем состояла в моделировании сложных быстропротекающих процессов и в получении достоверной информации об экспе-

риментальных результатах, получаемым по различным регистрациям проникающего излучения. Часть вопросов моделирования и развития численных методов обработки результатов экспериментов решалась в ИПМ АН СССР им М. В. Келдыша благодаря безупречной работе БЭСМ – 6.

Позвольте мне выразить глубокое уважение и признательность трудам академика В. А. Мельникова, а так же ряду руководителей и ученых, работавших в отрасли и непосредственно выполнявших исследования с помощью БЭСМ – 6. Это, прежде всего, Я. Б. Зельдович, Е. И. Забабахин, Л. П. Феоктистов, Е. Н. Аврорин. В МИФИ (1963-1969) А. Б. Мигдал, А. С. Компанец, В. И. Гольданский, В. Я. Арсенин, Д. А. Васильков. В НИИ-ИТ (1969-1991) В. Н. Михайлов, М. Н. Нечаев. В ИПМ АН СССР им М. В. Келдыша (1971- 1998) А. Н. Тихонов, В. Я. Арсенин, В. П. Загонов.



Е. И. Забабахин



В. Н. Михайлов



А. Н. Тихонов



В. Я. Арсенин



Я. Б. Зельдович



В. И. Гольданский

Ниже приводится краткий список работ [1 – 7] выполненных в ИПМ во время моей работы там с 1971 по 1994 г и послуживших основой методов обработки экспериментальных измерений. Некоторые из полученных результатов можно отнести к «истокам» ныне перспективных областей науки и техники таких, например, как управляемый термоядерный синтез, томография, мониторинг окружающей среды и др.

В качестве примера для краткой характеристики информации, использующейся при решении задач обработки экспериментов, остановимся на аннотации трех работ. Тема, затронутая в них – обработка результатов экспериментов [2, 3] и моделирование [5] по генерации и распространению электромагнитных импульсов (ЭМИ) в пространстве между Землей и Ионосферой. Данная тема продолжает оставаться актуальной и разрабатывается в настоящее время в различных академических институтах : ИРЭ РАН, ИЗМИР РАН, ИСЗФ СО РАН и др. . Интерес к теме связан с исследованиями как самих источников сигналов (например, грозových разрядов), так и каналов их распространения (например, Земной и/или Морской поверхности, Атмосферы и/или Ионосферы). Поскольку электромагнитные сигналы охватывают чрезвычайно широкий диапазон длин волн от 10^6 до 10^{-7} м (от КНЧ диапазона радиоволн до светового, рентгеновского диапазона), они являются уникальным средством диагностики в различных исследованиях, включая лазерную оптику [8] и квантовую электронику [9, 10].

В работе [3] выполненной непосредственно под руководством В. Я. Арсенина обрабатывалась серия экспериментальных результатов по регистрации ЭМИ, возбуждаемого импульсным источником гамма квантов, расположенным на земной поверхности. Регистрация сигнала $U(t, r)$ проводилась на различных удалениях r от источника с достаточно хорошо известными функциями передачи $K(t, r)$ наземных трасс распространения. Требовалось найти электромагнитный сигнал $U_0(t) = U(t, r = 0)$, генерируемый источником и неискаженный трассой распространения. Другими словами непосредственным результатом являлось решение «обратной» задачи – решение уравнения «свертки»:
$$U(t, r) = \int_0^t K(t - t^*, r) U_0(t^*) dt^*$$
 с неизвестной функцией $U_0(t)$.

Как известно данное уравнение – уравнение первого рода, при решении которого приходится сталкиваться с проблемами устойчивости и оценки точности решения в различных временных масштабах [2]. Кроме того, при обработке множе-

ства реализаций $U(t, r)$ стоит вопрос о фильтрации для устранения искажений полезного сигнала как от аппаратных «шумов», так и от случайных флуктуаций самих сигналов.

Особенность задачи состояла еще и в том, что на больших расстояниях от источника на импульс, распространяющийся вдоль земной поверхности, накладывались ионосферные отражения ЭМИ. Наиболее ранние по времени прихода сигналы относились к импульсам отраженным от нижнего D слоя ионосферы, более поздние – к отражениям от E и даже F слоев, расположенными на высотах более 80 – 100 км.

Благодаря совершенствованию методов решения подобных некорректно поставленных задач [1, 2]. и огромному объему проведенных в ИПМ РАН численных исследований удалось, на основе экспериментальных данных выделить импульсные функции отражений от нижней ионосферы (от D слоя) и идентифицировать отражения от E и F слоев. Оказалось, что регистрация ЭМИ в широком временном интервале и диапазонах радиочастот позволяет не только выделить специфические особенности источников сигналов, но и оценивать состояние ионосферы во время проведения экспериментов.

На основе полученных результатов разработаны различные методики регистрации и определения параметров нестационарных источников ЭМИ, успешно применявшиеся на практике. Иллюстрация решения «прямой» задачи – моделирование распространения радиоимпульсов вдоль земной поверхности при отражениях от двумерно неоднородного (по высоте и расстоянию r) нижнего D слоя ионосферы, приведено в работе [7].

В заключении позвольте поблагодарить оргкомитет конференции, посвященной 80-летию со дня рождения академика В. А. Мельникова. за предоставленную возможность участия.

Литература

1. В. Я. Арсенин, М. И. Пергамент, А. Х. Рахматуллина, А. Н. Тихонов «Постановка задач для разработки математических методов автоматической обработки экспериментальных данных, полученных оптическими способами диагностики высокотемпературной плазмы» Препринт ИАЭ им. И. В. Курчатова, ИПМ им. М. В. Келдыша АН СССР. М. 1973.
2. В. Я. Арсенин, В. П. Загонов, Р. А. Трахонитовская «О численном решении интегральных уравнений первого рода типа свертки на не-

- равномерных сетках» Препринт ИПМ им. М. В. Келдыша АН СССР №141, 1978.
3. В. Я. Арсенин, В. А. Горячев, В. П. Загонов, «О расчете импульсных функций трасс распространения с отражением от нижней ионосферы». Дифференциальные Уравнения, Т. 16, № 7, 1980 г.
 4. В. П. Загонов «Один подход к решению операторных уравнений с использованием априорной информации о производной» Препринт ИПМ им. М. В. Келдыша АН СССР №30, 1985.
 5. Г. Д. Васильков, В. А. Горячев, А. А. Думова «Внешние краевые условия в задаче расчета электромагнитных полей от точечного источника гамма – квантов» Препринт ИПМ им. М. В. Келдыша АН СССР №33, 1989.
 6. Г. Д. Васильков, В. А. Горячев, С. Г. Чубукова «Расчет электромагнитного сигнала от нестационарного гамма – источника в условиях барометрической неоднородности воздуха» Препринт ИПМ им. М. В. Келдыша АН СССР №32, 1991.
 7. В. А. Горячев, В. П. Загонов, С. В. Подоляко «Математическое моделирование распространения радиоимпульсов в условиях крупномасштабных возмущений ионосферы» Препринт ИПМ им. М. В. Келдыша РАН №112, 1994.
 8. S. M. Zakharov, V. A. Goryachev, “All-Optical Logic Gates for High-Speed Supercomputers. II. Design and Physical Peculiarities”, Laser Physics, v. 4, No. 1, pp. 1-22, 1994.
 9. V. A. Goryachev, S. M. Zakharov, “Dynamics of Transmission of Ultrashort Light Pulses by Thin-Film Cavity Structures”, Quantum Electronics, v. 27, No. 3, 245-248, 1997.
 10. V. A. Goryachev, S. M. Zakharov, “Dynamic Features of Interaction a Sequence of Ultrashort Laser Pulses and Planar Thin-Film Microcavities”, Journal of Experimental and Theoretical Physics, v. 87, No. 5, pp. 855-863, 1998.

Секция 1

Общая структура программного обеспечения тренажерного комплекса

Михайлюк М. В. , д. ф. м. н. , профессор, НИИСИ РАН

Торгашев М. А. , к. ф. м. н. , НИИСИ РАН

Хураськин И. А. , к. т. н. , НИИСИ РАН

(Работа выполняется при поддержке РФФИ, грант № 09-07-00149_a)

В докладе рассматриваются основные подсистемы тренажерного комплекса и возможные методы организации интерфейса между ними в случае их работы в одном потоке, в разных потоках на одном компьютере и на разных компьютерах. Кроме того, рассматривается проблема синхронизации вывода синтезируемых частей изображения при работе с несколькими визуальными каналами.

В целом тренажерный комплекс можно условно разделить на несколько подсистем: рабочее место оператора, в котором находятся реальные или виртуальные пульта управления; подсистема динамики, которая в каждый момент времени вычисляет новые значения динамических параметров объектов виртуальной сцены; подсистема визуализации, состоящая из одного или нескольких визуальных каналов (мониторов), на которые вы-

водятся изображения частей визуализируемой сцены. Часто эти подсистемы разрабатываются независимыми фирмами и встает задача объединения их в единый программный комплекс. Рассмотрим сначала возможный интерфейс подсистемы динамики и подсистемы визуализации, оформленных в виде динамических библиотек, при этом мы различаем три варианта работы этих подсистем.

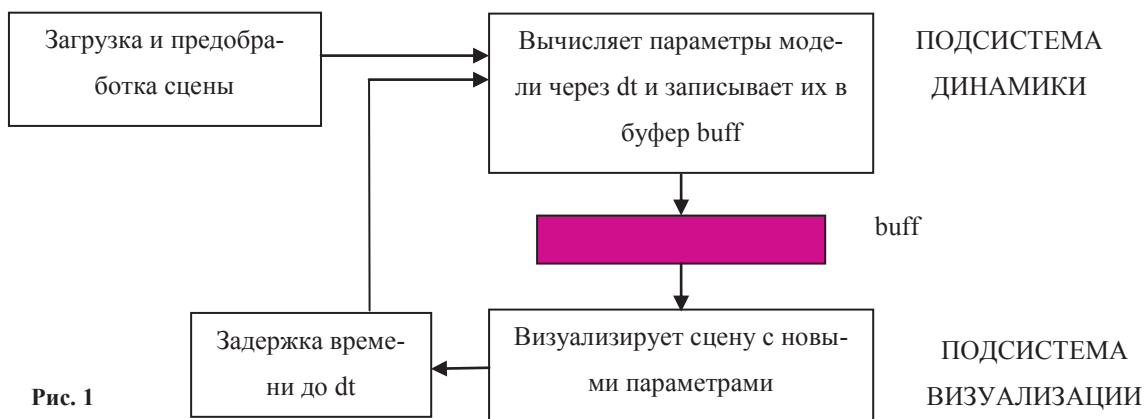


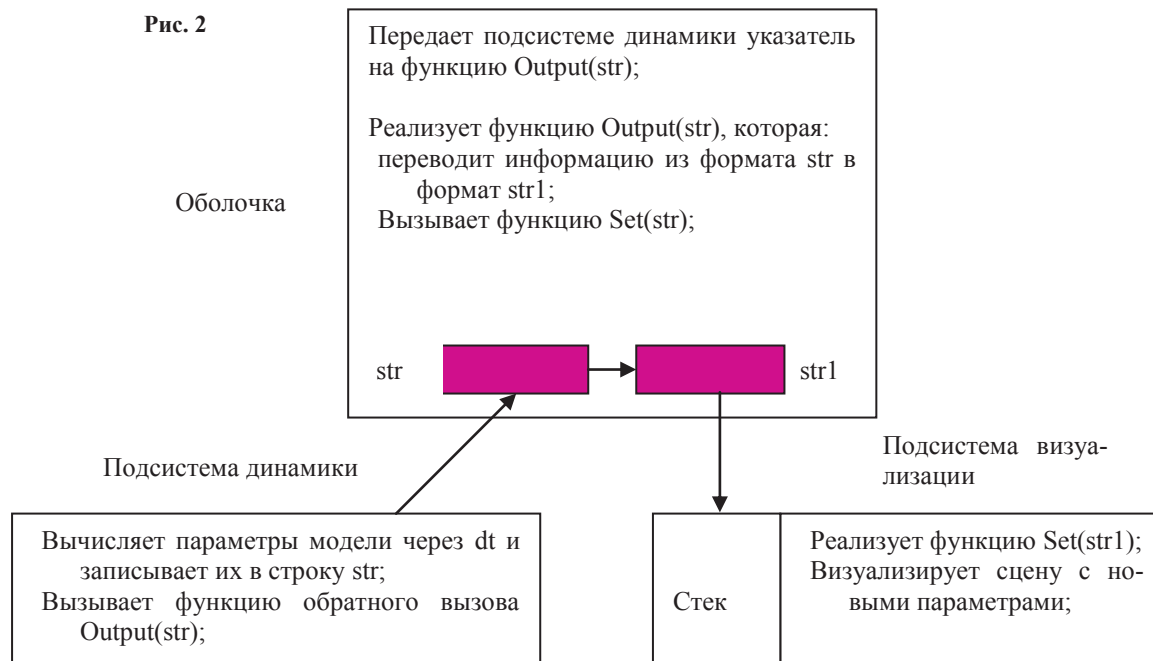
Рис. 1

Если подсистемы динамики и визуализации работают на одном компьютере в одном потоке (т. е. на одном процессоре), то их взаимодействие может быть организовано последовательно (см. рис. 1). Сначала подсистема моделирования рассчитывает динамические параметры модели через некоторый промежуток времени dt и запи-

сывает их в заданный буфер $buff$. После этого подсистема визуализации считывает эти параметры из буфера и визуализирует сцену с этими параметрами (например, с новыми положениями и ориентациями всех движущихся объектов). И, наконец, запускается модуль временной задержки, который сравнивает суммарное время расчета

динамики и визуализации с величиной dt . Если это время меньше dt , то процесс останавливается до тех пор, пока не наступит момент dt , а само значение dt уменьшается на некоторую величину. Если же больше, то dt увеличивается. После окончания работы модуля задержки управление опять передается подсистеме динамики и, таким образом, мы получаем рабочий цикл.

Если подсистемы динамики и визуализации работают на одном компьютере, но в разных потоках (например, на разных процессорах в многопроцессорных компьютерах), то возникает задача синхронизации работы этих подсистем.



Подсистема динамики может закончить вычисление параметров, в то время как система визуализации еще не закончила обработку текущего кадра. Ожидать этого окончания нет смысла, т. е. лучше это время потратить на расчет динамики для следующего кадра. В этом случае предлагается добавить в проект дополнительный модуль – оболочку, которая будет работать в собственном потоке и организовывать совместную работу этих подсистем (см. рис. 2). Эта работа будет проходить следующим образом. Подсистема динамики в своем потоке вычисляет новые значения динамических параметров через момент времени dt и записывает их в некоторую строку str . Затем она вызывает функцию $Output(str)$ обратного вызова, которая реализована в модуле оболочки. Эта функция переводит информацию из выходного формата подсистемы динамики во входной формат подсистемы визуализации (т. е. из строки str в строку $str1$) и вызывает функцию $Set(str)$. Функция $Set(str)$ реализована в подсистеме визуализации. Ее задача состоит в том, чтобы передать информацию строки $str1$ в подсистему

визуализации. Для того, чтобы обеспечить синхронизацию, эта функция переписывает строку $str1$ в стек. На этом заканчивается очередной цикл работы потока подсистемы динамики. Таким образом, в стеке накапливаются строки с динамическими параметрами до тех пор, пока подсистема визуализации не считывает их и не обрабатывает. Для корректной работы необходимо обеспечить целостность данных в стеке, исключив ситуацию одновременного чтения и записи с помощью одного из объектов синхронизации. Наиболее удобно использовать для этих целей критическую секцию (см. [1]). После окончания визуализации очередного кадра подсистема визуализации блокирует запись в стек с использованием критической секции, считывает из него все записи, все считанные параметры записывает в свои структуры (заменяя для каждого объекта значения его параметров более новыми), затем освобождает стек для записи и начинает визуализацию следующего кадра. Таким образом, каждая из подсистем работает независимо друг от друга, при этом не требуется переделывать каждую из

них для совместимости форматов (т. к. эту совместимость обеспечивает оболочка) и достигается приемлемая синхронизация их работы.

В третьем варианте подсистемы могут работать на разных компьютерах. В этом случае для связи предлагается использовать компьютерную сеть. Процесс работы в этом случае будет организован следующим образом. Подсистема динамики вычисляет динамические параметры модели, и каждые dt мсек (например, каждые 40 мсек) посылает подсистеме визуализации информационный пакет со значениями этих параметров. Для этого разработан специальный информационный протокол, а в качестве базового протокола передачи данных используется протокол UDP семейства TCP/IP. Подсистема визуализации получает пакет и визуализирует виртуальную сцену с новыми параметрами. При этом может возникнуть две ситуации. Первая – если подсистема динамики не успела провести вычисления или пакет не успел дойти по сети за время dt . Тогда подсистема визуализации производит экстраполяцию всех необходимых параметров на основе их предыдущих значений. Вторая ситуация – когда послано подряд несколько пакетов, а система визуализации еще не закончила формирование очередного кадра. Тогда пакеты будут накапливаться в стеке и будут считаны и обработаны после окончания визуализации кадра.

Проблему синхронизации нескольких каналов предлагается решать на основе технологии «клиент-сервер» (см. [2]). Для этого один из компьютеров (выделенный или обслуживающий какой-либо визуальный канал) выбирается в качестве сервера, на котором реализуется модуль синхронизации. Моделирующий комплекс передает вычисленные параметры модулю синхронизации (используя одну из вышеописанных технологий), а тот пересылает их всем компьютерам каналов (клиентам). Каждый канал на основе этих параметров визуализирует свою часть виртуальной сцены в заднем буфере кадра, однако не выводит изображение на экран (т. е. в лицевой буфер кадра). Вместо этого он посылает сигнал модулю синхронизации об окончании визуализации. Модуль синхронизации ожидает такие сообщения от всех компьютеров и, после получения последнего сообщения, посылает всем компьютерам команду на вывод сформированных изображений на экраны. Получив такие сообщения, клиенты производят смену своих задних и лицевых буферов кадра, и части изображения одновременно появляются на всех экранах. На рис. 3 показана схема, иллюстрирующая описанную технологию.

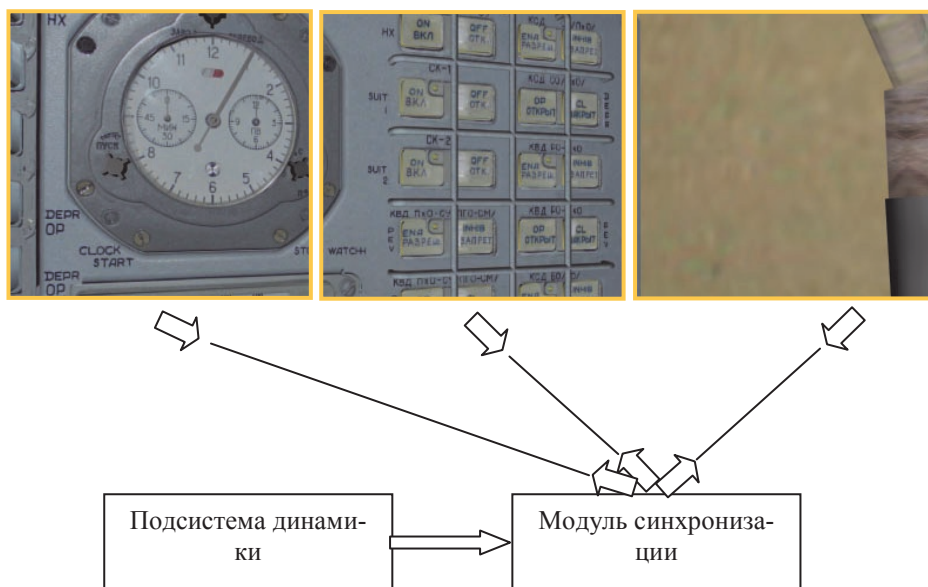


Рис. 3

Интерфейс пользователя, то есть связь подсистемы управления с подсистемами динамики и визуализации может осуществляться по такой же

схеме, как между этими подсистемами. Необходимо отметить, что управлять можно как отдельными объектами, так и камерами и источниками

освещения. При этом управляющие воздействия для объектов передаются подсистеме динамики, а для камер и источников освещения – подсистеме визуализации. В отладочном или нештатном режимах для управления можно использовать клавиатуру, мышь или джойстик. В штатном режиме используются виртуальные или реальные пульты управления совместно с подсистемой логики. Каждому элементу управления соответствует своя логическая схема. При воздействии оператора на этот элемент (например, переключение тумблера с помощью мыши), логическая схема вычисляет значение некоторого параметра (например, момента силы, подаваемой на двигатель), который затем передается подсистеме динамики по одной из описанных ранее схем. На рис. 4 показан пример виртуальной сцены с роботом и его виртуальным пультом управления.

В этом пульте с помощью мыши можно переключать все элементы (кнопки, тумблеры, регуляторы и др.), тем самым управляя действиями робота (его движением, перемещением манипулятора, захватом и т. д.).

Для начального задания положения объектов в сцене можно использовать конфигурационные скриптовые файлы. Это текстовые файлы, в которых на специально разработанном скриптовом языке записываются начальные значения параметров для объектов сцены. Файл обрабатывается после загрузки сцены на этапе ее предобработки. Кроме того, для тренажерных комплексов, использующих технологию виртуальной реальности, для интерфейса пользователя можно использовать компьютерные перчатки, системы трекинга и шлемы виртуальной реальности.



Рис. 4

Литература

1. Дж. Рихтер. Windows Для профессионалов, Microsoft Press, М. 2001.
2. М. В. Михайлюк, М. А. Торгашев, И. А. Хураськин. Система синхронизации синтеза и отображения виртуальных трехмерных сцен для распределенных имитационно-тренажерных комплексов. // Информационные технологии и вычислительные системы, № 4, 2003, стр. 48-55.

Физически корректное моделирование освещенности в задачах компьютерной графики

Волобой А. Г., к. ф.-м. н.

Институт прикладной математики им. М. В. Келдыша РАН
Работа выполняется при поддержке РФФИ (грант № 08-01-00649)

В статье представлен подход к синтезу реалистичных изображений, основанный на физически аккуратном моделировании распространения света. При этом подходе важной проблемой является задание корректных оптических свойств материалов и источников освещения виртуальной сцены. Рассмотрены методы и подходы к заданию этих начальных данных, необходимых для достижения правдоподобного результата моделирования и визуализации.

Введение

Методы синтеза реалистичных изображений, основанные на физически аккуратном моделировании распространения света в различных средах, являются одним из современных направлений развития компьютерной графики. Рис. 1 иллюстрирует этот процесс.



Рис. 1. Процесс генерации изображения.

Моделирование будем называть *физически аккуратным*, если методы моделирования основаны на физических законах распространения света и взаимодействия света с объектами. Под понятием *реалистичности* изображений будем понимать близость восприятия синтезированных изображений с восприятием фотографий, т. е. именно фотореалистичность является критерием качества синтеза изображений в нашем случае.

Созданные алгоритмы и программные средства представляют интерес для использования в архитектуре, градостроительстве, ландшафтном

дизайне, проектировании систем освещения и дизайне интерьеров, автомобильной и авиационной промышленности и др. Важным аспектом является не просто реалистичность получаемых изображений, а их соответствие законам физики и оптики. Именно это соответствие позволяет проводить моделирование освещения интерьеров, проверять нормы освещенности в процессе разработки проекта.

Одним из способов подтверждения правильности разработанных алгоритмов является проект, осуществленный совместно с университетом Aizu

Wakamatsu (Япония). Была воссоздана виртуальная модель одного из холлов университета. Были измерены и заданы максимально корректно оптические свойства материалов, использованных при строительстве этого холла, а также источники освещения, расположенные там. Далее виртуальная модель холла была рассчитана и визуализиро-



вана с помощью системы, разработанной в ИПМ. Сравнение результата моделирования с фотографиями реального холла показало хорошее соответствие. Небольшие отклонения в основном связаны с неточностью задания геометрии или индикатрис рассеяния источников освещения (рис. 2).



Рис. 2. Сравнение результатов компьютерного моделирования холла университета (рисунок слева) с реальной фотографией (рисунок справа).

Однако для получения правдоподобных результатов необходимо максимально корректно задавать входные данные: геометрию виртуальной сцены, оптические свойства всех материалов и источников света, используемых в моделировании. Проблема с точным заданием геометрии была решена с помощью разработки plugins к таким известным системам автоматизированного моделирования, как 3DS Max, популярного среди архитекторов и дизайнеров интерьеров, и Catia, широко используемой в автомобильной и авиационной промышленности.

Оптические свойства материалов

Правильное и физически корректное задание светорассеивающих (оптических) свойств изображаемых объектов и поверхностей является непростой задачей. В большинстве случаев эти свойства материалов неизвестны, и получить их можно только непосредственными измерениями. Наиболее общий способ описания светорассеивающих свойств поверхности – задание двунаправленных функций отражения/пропускания (ДФО/ДФП). ДФО/ДФП, заданная в каждой точке поверхности, определяет какая часть энергии, пришедшая с направления w_{in} , будет рассеяна поверхностью в направлении w_{out} . Таким образом, в общем случае, ДФО/ДФП в каждой точке является функцией

двух направлений. В ИПМ был разработан аппаратно-программный комплекс, предназначенный для измерения спектрального и пространственного распределения света, рассеиваемого или пропущенного поверхностями. Прибор основан на оригинальной концепции, обеспечивающей одновременное измерение рассеяния в различных направлениях и высокое пространственное разрешение в критической области вблизи направления зеркального отражения. Для параллельного получения большого объема пространственной информации используется камера на ПЗС-матрице.

Другим направлением стало моделирование новых оптически сложных материалов, таких как многослойные краски со сложной микроструктурой (типа перламутровых и «металлик»). В результате моделирования распространения света в их внутренней структуре, состоящей из плоскопараллельных однородных слоев, генерируется ДФО. Каждый слой состоит из прозрачного лака и взвешенных в нем пигментных частиц и интерференционных чешуек (рис. 3а). После проведенного моделирования полученные ДФО могут быть использованы при генерации реалистичных изображений автомобилей или других объектов, покрытых этими оптически сложными материалами (рис. 3б).

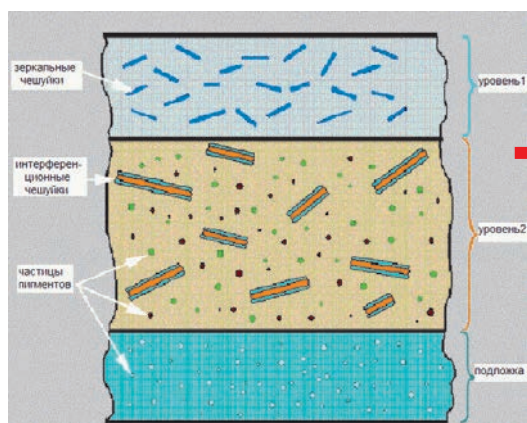


Рис. 3а. Структура многослойной краски.

Моделирование тонкого красящего слоя, как, например, являются чернила, характеризуется высокими концентрациями частиц и их маленькими размерами (порядка сотен нанометров). Лучевой подход или лучевое уравнение переноса практически не применимы для решения этой задачи. Поэтому был выбран путь непосредственного решения волновых уравнений, естественно, при некоторых ограничениях и упрощениях. В результате вычислительно-емкого моделирования распространения света в таком слое (время моделирования доходит до нескольких дней и даже недель на современных Core2Quad процессорах) получается ДФО, которая потом используется при генерации реалистичного изображения сцен, содержащих такие материалы.

Ткани являются сложным для визуализации и моделирования типом материала. Расчет ДФО ткани основан на моделировании освещенности. Освещенность ткани рассчитывается с учетом рассеяния света поверхностями и объемами нитей, ее образующих. Также принимается во внимание структура переплетения нитей, которая позволяет определить тени и свет, отбрасываемые одними нитями на другие. При расчете теней и объемного рассеяния света учитываются оптические характеристики материала, из которого состоит нить. Поверхностное рассеяние света нитью аппроксимируется по формуле Фонга. Полученный метод позволяет получать внешний вид тканей с учетом реальных оптических характеристик нитей разного типа (хлопковые, шелковые и др.).



Рис. 3б. Модель автомобиля, окрашенная смоделированной краской.

Задание источников освещения.

Для корректного задания источников света были разработана поддержка импорта гониограмм (индикатрис рассеяния) источников из формата IESNA. Большинство производителей осветительных приборов предоставляют эту информацию для своих продуктов.

Также было разработано моделирование естественного освещения. Небесное освещение задается с помощью географических координат, даты, времени суток и степени облачности. Моделирование небесного освещения для случаев 100% облачности и ее отсутствия соответствует стандарту Международной комиссии по освещению CIE. Небесное освещение с неполной облачностью моделируется с использованием интерполяции.

Другим методом задания освещения в графических комплексах может служить способ, при котором источником света является изображение реальных панорам с большим динамическим диапазоном яркостей. При этом источником света могут быть сфотографированные особым образом небо или помещение выставочного зала, и мы можем смоделировать, как будет освещен тот или другой объект в этих условиях. В результате использования этого подхода появляется возможность интегрировать моделируемые объекты с реальным окружением. На рис. 4 показана компьютерная модель автомобиля, «встроенная» в реальный природный ландшафт.



Рис. 4. Компьютерная модель автомобиля освещена сфотографированной панорамой

Заключение

Методы физически корректного моделирования распространения света показали себя эффективным инструментом в архитектурном проектировании, автомобильной, авиационной и электронной промышленности, а также при разработке новых оптически сложных материалов.

Были найдены и разработаны практичные и эффективные способы задания начальных данных – материалов и источников освещения, - необходимых для получения достоверных результатов моделирования.

Методика и алгоритмы визуализации для обучающих модулей компьютерных тренажерно-обучающих систем

Решетников В. Н. д. ф. -м. н. , проф. ЦВиСИТ НИИСИ РАН

Мамросенко К. А. ЦВиСИТ НИИСИ РАН

Работа выполняется при поддержке РФФИ (грант № 08-07-00025-а)

Тренировки космонавтов, операторов сложных технических систем на реальных установках и в реальных условиях слишком опасны и дороги, а часто не реализуемы. Альтернативой являются компьютерные имитационно-тренажерные комплексы, которые в максимально возможной степени моделируют реальные установки, порождают виртуальные объекты и позволяют тренирующимся приобрести правильные и устойчивые навыки.

В работе представлены: алгоритм построения программно-аппаратного комплекса, позволяющего в короткие сроки создавать интерактивные мультимедийные тренажерно-обучающие сессии, методика создания виртуальной окружающей среды, содержащей визуальную модель окружающего пространства, визуальные модели рассматриваемых объектов и явлений, разнородную аудиовизуальную информацию, методика построения визуального интерфейса управления виртуальным окружением, который позволяет оператору управлять системой и виртуальной средой, алгоритм построения модуля формирования структуры сессии.

Разработанные методика и алгоритмы могут быть использованы при построении распределенных тренажерно-обучающих систем различных проблемных направлений, обеспечивающих как индивидуальное, так и групповое обучение в условиях, максимально приближенных к реальности. Разработанные методика и алгоритмы также применимы при построении систем дистанционного и открытого образования.

Общие положения

Управление сложными техническими системами представляет сложный процесс, требующий длительного обучения и соответствующих тренировок. Подготовка операторов таких систем включает как теоретическую часть (изучение соответствующих математических моделей, теории игр, дифференциальных систем, критериев надежности и т. д.), так и практическую, без которой невозможно получение устойчивых навыков управления. Следовательно, возникает необходимость создания тренажерно-обучающих систем (ТОС), развития теорий и методов на базе технологий математического моделирования и виртуальной реальности.

Современные тренажерные системы с высокими требованиями к адекватности, обеспечивающие групповую тренировку, обычно строятся как совокупность завершенных подсистем моделирования, визуализации, ведения тренировки (запись, воспроизведение, фиксация и оценка) и пр.

Использование мультимедийных технологий при построении ТОС позволяет создавать банки данных учебных материалов, содержащие изображения, тексты, сопровождающиеся аудио, видео и другими визуальными эффектами; включают в себя интерактивный интерфейс; позволяют эффективно использовать накопленную информацию при организации дистанционного и распределенного обучения.

При создании ТОС, необходимо такое программное обеспечение, которое обеспечит оперативное объединение всех видов имеющейся учебной информации (текстовой, графической, мультимедийной, синтезированной и т. д.) в единое целое, которое можно передавать по различным каналам связи, использовать в задачах удаленной подготовки персонала.

На основе анализа требований к организации и изложению учебной информации создано программно-методическое обеспечение с использованием синтезированного окружения, максимально

приближенного к реальным условиям, с внедрением в виртуальное пространство графического образа инструктора. Предлагаемая архитектура программно-методического обеспечения состоит из трех основных взаимосвязанных частей: модуля подготовки структуры курса, программно-аппаратного комплекса подготовки и проведения обучения персонала и модуля структурированного отображения разнородного материала.

Модуль подготовки структуры курса

Построение структуры информационного наполнения тренажерно-обучающей системы является первым этапом подготовки сессии обучения и тренировки специалистов, базируется на текстовом описании курса подготовки. С целью оперативной сборки разнородных аудиовизуальных материалов в единое синтезированное (виртуальное) окружение, а также обеспечения компонентного построения структуры сессии разработан проблемно-ориентированный программный модуль. Модуль является загружаемым плагином для текстового процессора Microsoft Word и имеет графический интерфейс.

В диалоговой форме расположены доступные команды для визуализации основного текста курса. В зависимости от вида мультимедийной информации доступен выбор способа отображения в едином трехмерном окружении. По окончании выбора в тексте курса появляются специальные тэги, определяющие команды описания сценария отображения информации. Для передачи параметров в программно-аппаратный комплекс используются файлы сценария с собственной внутренней структурой, которые формируются путем извлечения тэгов из текста курса.

Как правило, загрузка высокодетализированных изображений в виртуальную трехмерную среду происходит с задержкой, что не допускается требованиями к быстродействию системы в целом. Для преодоления описанного эффекта был применен метод предварительной подготовки материалов. Данный метод заключается в предварительной загрузке разнородной информации, используемой в трехмерной среде, в память графического адаптера во время общей загрузки сцены. Программно метод был реализован путем извлечения команд из тегов курса, отвечающих за загрузку материалов и добавлением их в раздел файла сценария, предназначенного для загрузки материалов во время инициализации общей сцены.

Разработанный модуль позволяет существенно упростить сборку разнородных мультимедий-

ных материалов, значительно снизить количество времени на подготовку структуры, избавляя от необходимости вручную задавать команды на скриптовом языке, избегать появления ошибок при вводе данных.

Программно-аппаратный комплекс подготовки и проведения обучения персонала

На основе анализа требований к организации и изложению учебной информации создан программно-аппаратный комплекс (ПАК) на базе персонального компьютера с использованием синтезированного окружения, максимально приближенного к реальным условиям, с внедрением в виртуальное пространство графического образа инструктора. Таким образом, создается возможность проведения подготовки и тренировки обучаемого персонала с сохранением накопленного ранее опыта с возможностью создания и воссоздания нештатных ситуаций.

При создании ПАК главной задачей стала реализация нового метода организации учебной информации, который позволяет совмещать различные аудиовизуальные материалы в единое синтезированное окружение (проблемно-ориентированный виртуальный мир).

Особый интерес представляют следующие виды аудиовизуальной учебной информации: динамические графики развития процессов, диаграммы, гистограммы для анализа массивов данных, иллюстративные материалы изучаемых объектов различного происхождения, мнемосхемы, трехмерные модели объектов окружающей виртуальной среды и объектов исследования с применением математических моделей их описания и функционирования, функциональные схемы взаимодействия отдельных подсистем, а также обобщенные схемы работы изучаемой системы в целом, результаты работы программных продуктов в форме видео-образов с сохранением управляемости приложения, видеоматериалы реальных объектов.

Для создания виртуальной окружающей среды и моделей объекта используется синтез визуального изображения методами машинной графики с помощью систем компьютерной генерации изображений. Виртуальное окружение строится на основе обработки информации о трехмерной сцене, содержащейся в базе данных, с возможностью импорта данных в комплекс из различных источников. Для отображения информации используются моно- или стереорежим.

Для внедрения графических образов объектов реального мира в виртуальное окружение использовался метод цветовой электронной рир-проекции, реализованный в виде программного модуля и аппаратной платы расширения. Для практической реализации данного метода необходимы: студийное помещение с синей или зеленой циклорамой на заднике и синтезированная на ЭВМ окружающая обстановка, включающая в себя фон и объекты переднего плана.

ПАК выполняет основные задачи по созданию синтезированного виртуального окружения в реальном времени и строится на базе персонального компьютера с использованием операционной системы семейства Windows NT. Управление ПАК возможно осуществлять удаленно, используя локально-вычислительные сети, глобальную сеть Интернет и др.

Блок кеинга (рирпроекции) ПАК предназначен для выделения из видео-потока, получаемого с камер, изображения инструктора, а также дополнительных объектов реального окружения при помощи метода электронной цветовой рир-проекции.

Модуль визуализации ПАК предназначен для создания визуального образа из формального описания трехмерной сцены и внедрения выделенного графического образа инструктора, а также аудиовизуальной информации. Параметры объединения задаются в модуле управления виртуальным окружением.

Модуль управления виртуальным окружением ПАК позволяет оператору управлять системой и виртуальной сценой посредством визуального интерфейса. Интерфейс состоит из управляющих элементов, содержащих команды, написанные на специальном языке описания сценариев, результатом их выполнения является оперативная модификация виртуальной сцены.

В рамках исследований создано универсальное трехмерное окружение, позволяющее проводить тренировочные занятия с демонстрацией аудиовизуальной учебной информации.

Виртуальное трехмерное окружения содержит следующие компоненты:

- графический синтез математических моделей, описывающих процессы, реализуемые в виртуальной среде. Требования к составу, полноте и адекватности этих моделей определяются структурой курса.

- геометрические модели, обеспечивающие визуальное представление моделируемой среды либо на экране монитора, либо в стереосистеме с поляризационным экраном, двумя проекторами и

пассивными поляризационными очками. Геометрическая модель виртуального пространства формируется путем создания геометрических моделей всех объектов, которые помещаются в это пространство.

- модели акустической обстановки, обеспечивающие имитацию тех звуковых эффектов, которые сопровождают моделируемые события.

Для создания геометрических моделей в настоящее время имеется достаточно развитый арсенал как промышленных программных систем геометрического дизайна, так и специализированных разработок. Это – различные CAD-системы, пакет 3DS-MAX, языки семейства VRML и т. п.

Универсальное трехмерное окружение позволяет отображать аудиовизуальную информацию в трех различных формах: детальной, обобщенной, в форме видеоряда. В детальной форме инструктор может выделить отдельные части в рассматриваемой системе, подробно описывая каждую из них. Присутствует возможность отображения функциональных связей внутри системы, течения происходящих процессов.

В обобщенной форме информация разбивается на две части, совокупность которых образует полное описание поставленной задачи. Информация выводится на две независимые виртуальные панели и в зависимости от сценария курса меняется либо на обеих панелях, либо только на одной.

Модуль структурированного отображения разнородного материала

Данный модуль является инструментальным средством проведения подготовки персонала. В него поступают: информация из программно-аппаратного комплекса, текстовое описание процесса подготовки, трехмерные и двумерные модели рассматриваемого объекта и среды функционирования, набор правил для проведения тренировки.

Модуль построен на базе технологии структурированного отображения разнородного материала, заключающейся в демонстрации подготовленного мультимедийного материала в структурированном виде с привязкой к текстовой информации курса. Эта технология также обеспечивает интерактивность за счет модуля трехмерной визуализации и интерактивной двумерной анимации.

Схема отображения информации создается на основе текста курса в формате RTF, в котором формируется иерархическая структура (дерево)

пунктов, каждому из которых сопоставляется место в общем описании, а также некоторая визуальная информация, наиболее эффективная для описания данного пункта. При просмотре обучающей сессии пользователь может в произвольном порядке перемещаться по дереву пунктов, при этом в окнах визуального интерфейса отображается соответствующая информация из общего текстового описания и сопоставленная ему визуальная информация.

В системе предусмотрена возможность поиска терминов или фраз, как по основному тексту курса, так и по иерархическому меню.

В целях более детальной проработки особо важных вопросов, в текст курса внедряются гипертекстовые ссылки на аудиовизуальные материалы, сформированные ПАК, где подробно описан интересующий вопрос или термин.

Литература

1. В. Н. Решетников, М. А. Торгашев, И. А. Хураськин. Система создания и просмотра мультимедийных инструкций. // Программные продукты и системы – 2007, № 2.
2. М. В. Михайлюк, В. Н. Решетников «Имитационно-тренажерные и обучающие распределенные системы», «Сборник трудов НИИСИ РАН», 2007 год.
3. Б. О. Любимов, Г. А. Михальков, В. Н. Решетников Система отображения ситуационного центра, Труды Научной конференции посвященной 75-летию со дня рождения академика В. А. Мельникова –М. : Типография Президиума РАН, 2004 г.

Архитектура ускорителя трехмерной графики для ЭВМ серии Багет

Евлампиев Б. Е. , Корниленко А. В. , Костюхин К. А.

НИИСИ РАН

1. Постановка задачи

- В настоящее время интенсивно развиваются геоинформационные системы, они получили широкое распространение в различных областях промышленности. Это делает актуальным разработку графических контроллеров (ГК), способных работать в жестких условиях окружающей среды и имеющих производительность, достаточную для решения задач картографии и навигации. Проведен обзор структур современных графических контроллеров, на основании которого выделены основные архитектуры:
- многоядерные потоковые процессоры для персональных компьютеров и игровых приставок, ориентированные на получение максимальной производительности в электронных играх при обработке трехмерных сцен и рассчитанные на диапазон температур $0 \div +60$ °С.
- графические контроллеры, предназначенные для применения в жестких условиях эксплуатации (при температуре $-40 \div +85$ °С) для обработки двухмерных и трехмерных сцен, вывода изображения на экраны размером не более 1024×768 пикселей и работающие под управлением специализированных процессоров;
- графические контроллеры для мобильных применений, рассчитанные на работу в диапазоне температур $0 \div +60$ °С, которые обеспечивают получение производительности около 15 кадров/с на экране размером не более 512×512 пикселей при уровне энергопотребления около 0,3 Вт.
- Проведенный обзор показывает, что в настоящее время отсутствуют контроллеры, предназначенные для работы в жестких условиях окружающей среды и способные формировать изображение размером 1600×1200 пикселей, а также работать с двухмерными и трехмерными объектами.

2. Архитектура ускорителя

Для реализации этой задачи предлагается архитектура контроллера, блок-схема которого представлена на рис. 1.

Основными блоками этого контроллера являются: интерфейс центрального процессора, блок обработки вершин, блок наложения текстур, блок управления 3D, контроллер памяти и коммутатор.

В качестве интерфейса центрального процессора используется шина AXI. Этот интерфейс используется для записи/чтения регистров и видеопамати. Выбор шины AXI объясняется следующими причинами. Первое, эта шина является одним из распространенных стандартов для объединения IP блоков на кристалле. Второе, в НИИСИ РАН накоплен богатый опыт использования этой шины и создан набор IP блоков на ее основе. Это позволяет, как легко интегрировать данный контроллер в состав другой микросхемы, так и строить на ее основе отдельное устройство. В последнем случае для построения микросхемы необходимо использовать отдельный блок, выполняющий функции моста между шиной AXI и какой-либо внешней шиной микросхемы. Для построения контроллера трехмерной графики использована шина AXI с шириной шины данных 64 разряда и максимальной частотой работы 166 МГц.

Блок обработки вершин предназначен для выполнения целого ряда задач, среди которых необходимо отметить следующие:

- преобразования координат вершин треугольников;
- расчет освещенности вершин треугольников;
- отсечение поверхностей, лежащих за пределами видимой области экрана;
- расчет координат текстур.

Для построения этого блока необходимо использования SIMD ядра. Среди основных требований к этому ядру необходимо отметить следующие:

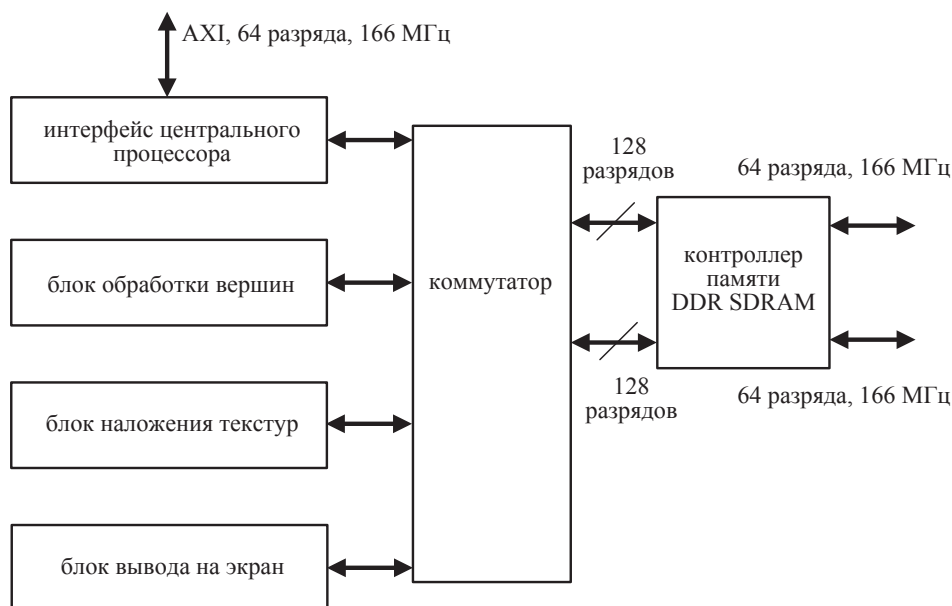


Рис. 1. Блок-схема контроллера трехмерной графики.

- работа с 32-разрядными числами в формате с плавающей запятой;
- перемножение матриц размерностью 4×4 ;
- расчет обратной величины;
- расчет квадратного корня от обратной величины.

Блок наложения текстур предназначен для расчета цвета пикселя на основе информации об освещенности вершин, параметра глубины вершин и координат текстуры. Также возможно использование различных методов смешивания текстур. Для этого блока предъявляются следующие основные требования:

- эффективное использование кэш памяти для снижения требований по пропускной способности шины видеопамати;
- одновременное наложение текстур на несколько пикселей;
- реализация проверки параметра глубины пикселя на ранних стадиях конвейера для исключения обработки невидимых на экране пикселей.

Блок управления 3D предназначен для управления блоками обработки вершин и наложения текстур.

Коммутатор необходим для реализации обмена данными между различными блоками. Фактически он реализован на основе нескольких 128-разрядных шин соединяющих четыре блока. Максимальная частота его работы 166 МГц.

Контроллер памяти необходим для управления видеопаматью. Предполагается использование двухканального контроллера памяти - две 64-разрядных шины с отдельным управлением, мак-

симальная частота работы 166 МГц (по двум фронтам). Контроллер будет работать в двух режимах: раздельная память, единая память. Первый режим, один канал памяти используется для хранения пиксельных данных, второй канал используется для хранения текстур и информации о параметре глубины пикселей. В этом случае наиболее интенсивный обмен данными будет возложен на второй канал памяти. Второй режим, управление видеопаматью будет реализован таким образом, чтобы сократить накладные расходы на переключение страниц в динамической памяти.

3. Блок наложения текстур

Текстурный конвейер состоит из последовательно выполняемых стадий. На первом этапе происходит выборка данных из области видеопамати, отведённой под хранение текстур. После получения данных происходит распаковка сжатых текстур.

Суть технологии в том, что текстура разбивается на квадраты 4 на 4 точки. Затем формируется таблица выборки текселей, при которых все цвета сводятся к 4-м, помещённым в данную таблицу, и каждому из 16 текселей присваивается двухразрядный код, соответствующий таблице выборки. Таким образом, данная система сжатия текстур имеет простой декодер, а кроме того, поскольку текстура разбивается на независимые квадраты, то возможна параллельная распаковка текстуры. В предлагаемом текстурном блоке возможна од-



Рис. 2. Блок-схема блока наложения текстур.

новременная распаковка четырёх сжатых 16-тексельных участков текстуры.

Далее для треугольника определяется, какие именно пиксели принадлежат треугольнику. С помощью алгоритма Брезенхема по вершинам строятся три прямые линии, ограничивающие область. Суть метода в следующем: для генерирования отрезков прямых Брезенхем на каждом шаге использует точку, наиболее близко расположенную к идеальному отрезку.

Затем каждой точке треугольника присваивается соответствующий текстель текстуры, исходя из заданного смещения текстуры относительно

вершины треугольника. После получения вида треугольника с нанесённой текстурой производится преобразование координат каждой точки на плоскость экрана. Для этого существуют три типа матриц: поворота вокруг оси абсцисс, ординат и аппликата соответственно. Кроме того, могут использоваться матрицы масштабирования и линейного переноса.

Последним этапом работы текстурного блока является фильтрация изображения. Фильтрация может применяться как билинейная, так и трилинейная, так и технология mip-текстурирования. Структура получившегося текстурного блока изображена на рисунке 2.

Умножение матриц производится с фиксированной запятой. При преобразовании координат текстелей текстур не требуется производить умножение с плавающей запятой, так как точности фиксированной запятой достаточно для преобразования текстуры. Кроме того, за счёт фиксированного значения порядка и отсутствия знака, точнее, только положительного его значения, умножение фактически превращается в простое целочисленное и может быть выполнено за один такт.

Значения тригонометрических функций вычисляются процессором, который заполняет матрицу, то есть прописывает регистры, соответствующие строкам матрицы.

4. Аппаратные средства профилирования

Современные процессоры общего назначения, такие как процессоры семейства x86, IA-64, Power, MIPS и др., поддерживают профилирование программ посредством использования аппаратных счетчиков – специальных регистров, фиксирующих аппаратные события определенного типа.

В качестве примера аппаратных событий можно привести общее число процессорных тактов, общее число выполненных команд, число выполненных операций с плавающей точкой, число промахов при обращении к виртуальной и кэш-памяти и др.

На сегодняшний день для разработчиков ПО, осуществляющего профилирование кода, стандартом de facto является проект PAPI (Performance Application Programming Interface). На рис. 3 представлена архитектура PAPI.

В PAPI выделяется два основных уровня: платформо-независимый и платформо-зависимый, скрывающий от пользователя детали реализации доступа к аппаратным счетчикам конкретного процессора. Для этого, в частности,



Рис. 3. Архитектура РАРІ.

функции РАРІ платформо-зависимого уровня могут использовать расширения ядра, функции целевой операционной системы или непосредственно язык ассемблера.

В состав высокоуровневого интерфейса входит всего 7 функций, предоставляющих базовые операции над аппаратными счетчиками: запуск, останов, чтение со сбросом и без. При этом можно использовать только события, предопределенные стандартом РАРІ.

Функции интерфейса верхнего уровня используют низкоуровневый интерфейс РАРІ, освобождая пользователя от явных вызовов, например, функции инициализации библиотеки РАРІ.

Низкоуровневый интерфейс обладает по сравнению с высокоуровневым интерфейсом расширенной функциональностью и большей эффективностью. В его состав входит более 50 различных функций, которые можно условно разделить на следующие группы:

- инициализация библиотеки РАРІ,
- функции измерения времени,
- функции получения информации,
- служебные функции,
- функции управления множествами событий,
- функции управления аппаратными счетчиками.

РАРІ обладает рядом дополнительных свойств: мультиплексирование, области действия счетчиков событий, переполнение счетчиков событий.

Мультиплексирование позволяет привязывать к одному аппаратному счетчику несколько событий. В этом случае за счет некоторой потери точности и дополнительных расходов, связанных с программным сохранением и восстановлением содержимого счетчика, появляется возможность

более широкого профилирования в условиях дефицита аппаратных ресурсов.

В РАРІ определяются следующие области действия счетчиков событий:

- подсчет событий производится, только когда процессор выполняется в режиме пользователя,
- подсчет событий производится, только когда процессор выполняется в режиме ядра,
- подсчет событий производится, только когда происходит обработка исключительных ситуаций, включает в себя все предыдущие области.

Переполнение возникает в том случае, когда число возникших событий превышает максимальное число, способное поместиться в регистре-счетчике. В этом случае РАРІ предусматривает возможность выполнения специальных обработчиков, задаваемых пользователем.

Для GPU предлагается добавить несколько аппаратных событий, а именно:

- произошла выборка текстуры из кэша текстур;
- произошла выборка блока данных из видеопамати.

Текстурирование больших регулярных сеток виртуальных сцен в имитационно-тренажерных комплексах

Тимохин П. Ю. (ЦВуСИТ НИИСИ РАН)

(Работа выполняется в рамках проекта 2.5 ПФИ ОНИТ)

В данной статье решается проблема наложения множества уникальных фрагментов текстуры на полигональную модель, полученную из регулярной сетки вершин. В ходе решения задачи выбирается способ передачи данных графическому процессору, обеспечивающий высокую скорость визуализации, разрабатывается метод организации и выборки текстурных координат вершин. Для минимизации общего объема используемых вершинных данных предлагается подход, основанный на раздельном хранении атрибутов вершин. В заключение демонстрируется реализация метода в программном модуле, визуализирующем поверхность Земли в режиме реального времени.

В настоящее время в современных имитационно-тренажерных комплексах визуальная обстановка моделируется путем отображения виртуальной сцены в режиме реального времени. В ряде задач, например, при моделировании поверхности Земли или ландшафта местности, виртуальная сцена может содержать трехмерный объект, заданный регулярной или эквивалентной ей сеткой вершин размером $M \times N$ квадов, где квад – четырехугольник, заданный отдельной группой из 4-х вершин. Под вершиной понимается точка в трехмерном пространстве, обладающая определенным набором атрибутов (позиция, текстурные координаты и др.). Для получения реалистичной трехмерной модели в реальном времени берется сетка с большим количеством вершин, по ней создается полигональная модель, на которую затем накладывается текстура с высоким разрешением. Размер такой текстуры обычно исчисляется гигабайтами, что не позволяет поместить ее целиком в локальную память графического ускорителя и использовать для закраски модели. По этой причине текстура представляется в виде множества уникальных фрагментов, которые подкачиваются в видеопамять в процессе визуализации и накладываются на отдельные участки полигональной модели в режиме реального времени. При этом, как показано на рис. 1, текстурные координаты вершины, лежащей на границе смежных текстурных фрагментов, будут задаваться для каждого фрагмента в его системе координат: для нижнего

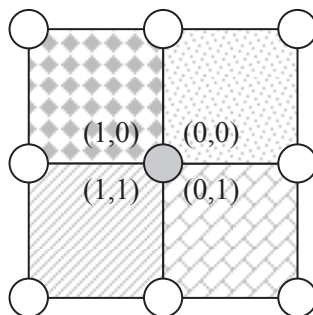


Рис. 1.

левого – (1, 1), для нижнего правого – (0, 1) и т. п. Поскольку для закраски полигона цвет в его вершинах должен быть однозначно определен, возникает задача выбора необходимых текстурных координат вершины в зависимости от того, в

визуализации какого текстурного фрагмента она задействована. Чтобы решить эту задачу, обеспечив высокую скорость визуализации, необходимо определиться со способом передачи атрибутов вершин графическому процессору (GPU).

Передача атрибутов вершин в GPU.

Применительно к графической библиотеке OpenGL можно выделить следующие основные способы передачи атрибутов вершин (вершинных данных) графическому процессору:

- передача атрибутов из оперативной памяти для каждой вершины отдельно (команды `glTexCoord()`, `glNormal()`, `glVertex()` и др.);
- передача массива атрибутов вершин (Vertex Arrays, VA) из оперативной памяти (команды `glTexCoordPointer()`, `glNormalPointer()`, `glVertexPointer()` и `glDrawArrays()` или `glDrawElements()`);

- передача массива атрибутов вершин из локальной видеопамати (расширения OpenGL Vertex Buffer Objects (VBO), Vertex Array Range (VAR) и др.).

Передача данных для каждой отдельной вершины ограничивает эффективность визуализации при большом числе вершин из-за частого обращения к GPU в каждом кадре [2]. Использование вершинных массивов (второй способ) снижает частоту обращений к GPU, однако, как и в первом случае, передача осуществляется по медленной шине CPU – GPU (пропускная способность шины AGP 8x - 2,1 Гб/с, PCI-E 16x - 4 Гб/с). Описанные выше ограничения можно обойти (используя третий способ), если загрузить на этапе предобработки все атрибуты вершин в буферы локальной памяти графического ускорителя и обновлять их лишь в случае необходимости [3]. При таком подходе GPU получает нужные вершинные данные очень быстро (пропускная способность видеопамати GeForce 8800 GTX - 86,4 Гб/с, Radeon X1950 XTX - 64 Гб/с) и скорость визуализации значительно возрастает (для технологии VBO приблизительно в 5 -10 раз).

В полигональной модели, построенной по сетке вершин, каждая вершина принадлежит нескольким графическим примитивам, поэтому во избежание дублирования всех ее атрибутов вводится индексирование вершин (рис. 2.) и дубли-

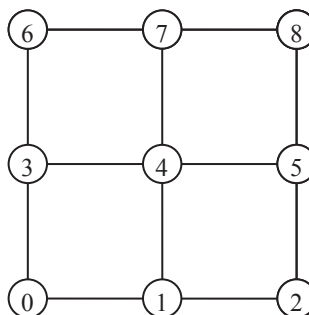


Рис. 2.

руются только индексы. Полученный массив индексов вершин также загружается в буфер видеопамати, называемый индексным буфером.

В концепции VBO [3] работа с видеопаматью осуществляется с помо-

щью относительной адресации, т. е. путем считывания определенного количества данных с заданным шагом и смещением относительно начала буфера, поэтому описанная выше задача по выбору текстурных координат вершины, сводится к разработке механизма управления смещением текстурных координат в буфере вершинных данных.

Расчет смещения текстурных координат вершин.

Пусть задана полигональная сетка $M \times N$ квадов, такая, что на каждый квад накладывается уникальный текстурный фрагмент. Принцип хранения и доступа к индексам и атрибутам вершин этой сетки, расположенным в локальной видеопамати, показан на рис. 3.

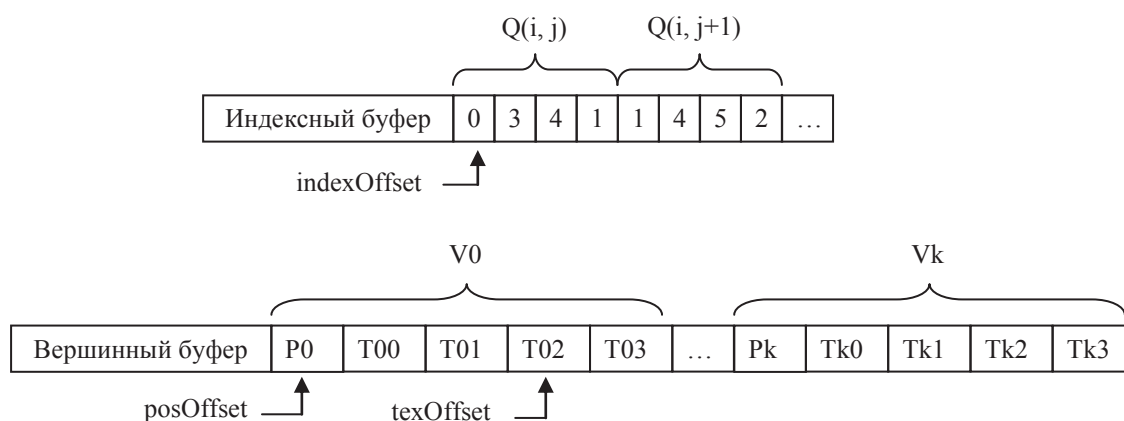


Рис. 3.

Для визуализации квада $Q(i, j)$ на CPU необходимо задать смещение $indexOffset$ от начала индексного буфера, задающее индекс первой вершины V квада, и количество считываемых индексов (в данном случае 4). Также на CPU необходимо вычислить параметры $posOffset$ и $texOffset$, опре-

деляющие соответственно смещения позиции P и текстурных координат T относительно смещения начала атрибутов вершины V квада $Q(i, j)$ в буфере вершинных данных. По индексам вершин квада GPU автоматически считывает из вершинного

буфера необходимые позиции вершин и текстурные координаты согласно формулам:

$$P[\text{index}] = \text{index} \cdot \text{sizeofVertex} + \text{posOffset},$$

$$T[\text{index}] = \text{index} \cdot \text{sizeofVertex} + \text{texOffset},$$

где sizeofVertex – число байт памяти, отводимое под атрибуты одной вершины.

Смещение texOffset рассчитывается на основе четности номеров строки и столбца квадрата в сетке вершин. Для этого номера берутся по модулю 2, и по ним вычисляется смещение $\tau \in \{0,1,2,3\}$ в одномерном массиве из 4-х вариантов текстурных координат вершины:

$$\tau = ((i \% 2) \cdot 2 + (j \% 2)) \cdot \text{sizeofTexCoords},$$

где sizeofTexCoords – число байт памяти, отводимое под одну пару текстурных координат.

Итоговое смещение текстурных координат для всех вершин каждого из 4-х смежных квадратов будет иметь вид:

$$\text{texOffset} = \tau + \text{firstTexOffset},$$

где firstTexOffset – смещение в байтах начала массива текстурных координат относительно начала атрибутов вершины.

Уменьшение избыточности вершинных данных.

Если посмотреть на сетку квадратов с рассчитанными смещениями τ (рис. 4), то видно, что значения текстурных координат вершин квадрата, взятые с этим смещением будут повторяться через строку и через столбец. Для уменьшения избыточности можно разделить вершинный буфер на буфер позиций вершин и буфер текстурных координат, а

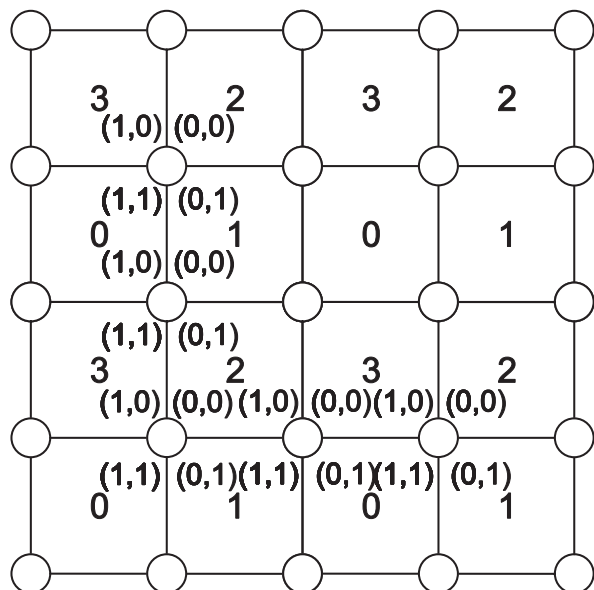


Рис. 4.

затем разбить общий буфер позиций вершин на p буферов, содержащих равное количество столбцов ветки вершин, а буферы текстурных координат и индексов вершин сократить в p раз. При этом общее количество позиций вершин будет увеличиваться на $p-1$ столбцов сетки вершин (т. к. позиции столбцов разбиения придется дублировать), один буфер текстурных координат будет соответствовать каждому из p буферов позиций вершин.

Параметр p выбирается таким образом, чтобы общий объем памяти был минимален. Введем обозначения $n = N + 1$, $m = M + 1$. При делении текстурный буфер будет иметь $(n-1)/p$ столбцов квадратов, т. е. $(n-1)/p + 1$ столбцов вершин. Так как каждый столбец содержит m вершин, каждая из которых включает 3 геометрические координаты и 4 пары текстурных координат, то общий объем необходимой памяти будет выражаться формулой:

$$V(p) = 3 \cdot (m \cdot n + m \cdot (p-1)) + 8 \cdot m \cdot \left(\frac{n-1}{p} + 1 \right)$$

Вычислив и приравняв к нулю производную функции $V(p)$, находим, что $V(p)$ будет минимальна при

$$p = \sqrt{\frac{8 \cdot (n-1)}{3}}.$$

В качестве окончательного из двух ближайших к p целых чисел надо выбрать то, для которого $V(p)$ имеет меньшее значение.

Аналогично можно сократить количество избыточных вершинных данных, разделив буфер позиций на p' буферов, содержащих равное количество строк сетки вершин.

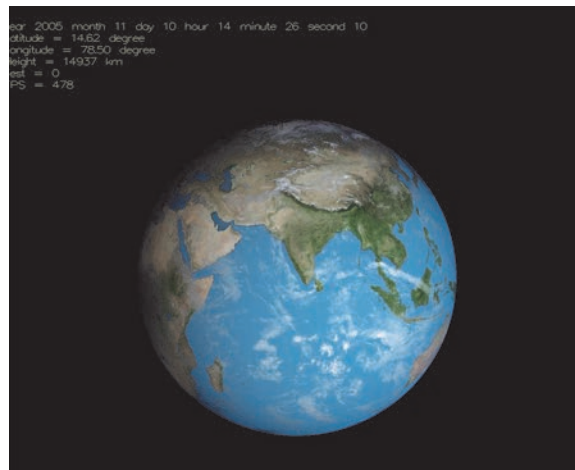


Рис. 5(1).

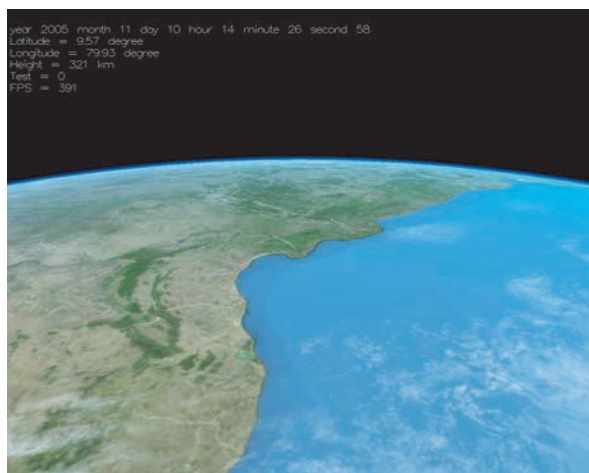


Рис. 5(2).

Пример программной реализации.

Описанная технология текстурирования больших регулярных сеток была реализована в программном модуле, визуализирующем поверхность Земли в режиме реального времени (рис. 5). Модель Земли представлена полигональной сферой, построенной из сетки вершин, эквивалентной регулярной, в которой число строк – это число параллелей, а число столбцов – число меридианов

на сфере. Размер сетки составляет $97 \cdot 193 = 18721$ вершин. Для уменьшения избыточности хранения вершинных данных использовалось $p = 6$ буферов позиций вершин. Разрешение текстурных фрагментов составляет 512×512 пикселей при исходном разрешении текстуры 24576×49152 пикселей.

Список использованных информационных источников.

1. Михайлюк М. В. Основы компьютерной графики. – М. : «МАТИ» – РГТУ им. К. Э. Циолковского, 2002.
2. Matthias Wloka, “Batch, Batch, Batch: What does it really mean?”, NVIDIA Corporation, GameDevelopers Conference, 2003.
3. Using Vertex Buffer Objects (VBOs), NVIDIA Corporation, 2003.
4. Технологии эффективного рендеринга геометрии. <http://www.gamedev.ru/articles/?id=20124>

Трехмерное динамическое моделирование работ по реабилитации пункта временного хранения ОЯТ и РАО в пос. Гремиха

Степеннов Д. Б., Максимов С. Б., РИЦ «Курчатовский институт»

Для реализации проекта по экологической реабилитации пункта временного хранения (ПВХ) ОЯТ и РАО в п. Гремиха в РИЦ «Курчатовский институт» разработана информационно-аналитическая система (ИАС), аккумулирующая в себе информацию по проекту и обеспечивающая возможность решения различных задач, возникающих перед проектной командой в процессе выполнения на всех стадиях работ по проекту.

Данные ИАС используются, в том числе, для проведения моделирования работ по проекту. В состав ИАС входит трехмерная модель ПВХ, включающая в себя модели рельефа местности, дорог, зданий и сооружений, объектов реабилитации и технического оборудования, модель акватории Святоносского залива. Эти данные в полном объеме используются для моделирования технологических схем обращения с радиоактивными отходами.

1. Изготовление трехмерной модели

Трехмерная модель территории изначально была изготовлена на основе планов территории 60-х годов. Затем, в связи с тем, что планы содержа-

ли большое количество расхождений с текущим состоянием объекта, в 2006 году было проведено геодезическое исследование, по итогам которого трехмерная модель ПВХ была уточнена.



Трехмерная модель ПВХ в п. Гремиха, прилегающая территория и акватория

Общая последовательность действий сценария моделирования была упрощена исходя из принципиальности каждого действия и доступных возможностей среды моделирования, в качестве которой был выбран программный продукт Walkinside VRContext.

Сценарий моделирования включает в себя следующие этапы:

- Установка краном контейнера на транспортное средство.
- Транспортировка контейнера по заданному маршруту.
- Перегрузка краном из транспортного средства в шахту поста перегрузки.
- Выгрузка отработавшей тепловыделяющей сборки (ОТВС) из контейнера.
- Загрузка ОТВС в транспортный контейнер на борт судна для вывоза с ПВХ.

Для реализации сценария моделирования были разработаны модели техники и оборудования, участвующие в работах: автомобильный стреловой кран КС-55713-1, автомобиль КАМАЗ 65115-62 с УКТС, пост перегрузки ОТВС из контейнеров, перегрузочный контейнер Сб. 02 ОК-300ПБ, Транспортный контейнер типа 18 (ТК-18), траверса для ТК-18, кран порталный монтажный КПМ-10-30-10ВП, передвижная эстакада мостового крана МК-75/20, судно т/х «Серебрянка».

На основе транспортно-технологической схемы был создан обучающий видеофильм, демонстрирующий процесс перегрузки ОТВС. Для создания видеофильма использовалась технология моделирования отдельных операций процесса в среде WalkInside и последующего сбора фрагментов в единый видеоряд согласно хронологической последовательности.

Выводы

Трехмерная визуализация и моделирование позволяют решить следующие задачи:

- Минимизация дозовой нагрузки на персонал при проведении работ в соответствии с принципом ALARA за счет предварительного трехмерного моделирования и выбора оптимальных технических решений.
- Составление сценариев на основе разработанных последовательностей технологических операций с последующей визуализацией с целью моделирования поведения персонала при исполнении работ и в случае возникновения чрезвычайных ситуаций.
- Просмотр описательной информации из базы данных ИАС непосредственно в процессе работы

с реалистичными трехмерными моделями ПВХ, зданий, сооружений, объектов в режиме реального времени.

- Создание демонстрационного материала на основе разработанных сценариев с целью разъяснения их сути исполнителям, контролирующим органам и заинтересованным лицам. Это позволит уменьшить сроки согласования проектных решений в надзорных органах.

Литература

1. Степеннов Д. Б., Беляева Н. М., Максимов С. Б., Метейко Д. А., Долгов Н. А., Орлик С. В. Применение аппаратно-программного комплекса ИАС для ввода и обработки данных комплексного инженерного и радиационного обследования пункта временного хранения ОЯТ и РАО в п. Гремиха // Сборник докладов отраслевого семинара «Аппаратурное обеспечение ядерной и радиационной безопасности объектов Росатома» (Ядерное приборостроение-2007) Россия, г. Москва, ФГУП «НИЦ «СНИИП» 18 – 19 апреля 2007 г. Стр. 93-102
2. Быстрова О. С., Шелованов Д. В. Разработка механизмов визуализации данных радиационных обследований ПВХ ОЯТ и РАО в п. Гремиха при помощи технологий трехмерной визуализации // Сборник трудов 5-й Курчатовской молодежной научной школы. 19-21 ноября 2007 г. Москва. РНЦ «Курчатовский институт». Стр. 100-101
3. Степеннов Д. Б., Максимов С. Б. «Информационно-аналитическая система сопровождения работ по реабилитации ПВХ ОЯТ и РАО в п. Гремиха». // Каталог инновационных проектов в области обращения с РАО, вывода из эксплуатации и экологической реабилитации. 2007 г., Росатом. Стр. 156.

Система отслеживания движений человека на основе технологии беспроводных сенсорных сетей для визуализации аватара

Вабищевич А. Н. , магистрант, Панфилов П. Б. , проф. , к. т. н. , (МИЭМ, г. Москва)

Описывается разрабатываемая система отслеживания движений человека на основе технологии беспроводных сенсорных сетей (БСС) для визуализации виртуального двойника-аватара. Дается краткое описание технологии БСС. Приводятся технические характеристики аппаратной части системы. Описывается программная студия для визуализации отслеживаемых движений в реальном времени. Рассматриваются возможные будущие приложения системы.

Введение

В настоящее время системы отслеживания движений активно используются в киноиндустрии, рекламе, анимации, виртуальной реальности, робототехнике и т. д. В основном такие системы обладают высокой стоимостью, для их применения требуется дополнительное оборудование и специальные площадки. Эти недостатки не позволяют использовать данные системы в повседневной жизни и в расширенном круге задач.

Технология БСС относительно молодая технология. В основном она широко применяется для различного стандартного мониторинга (климат-контроль, системы «умный дом», охранный мониторинг, пожарный мониторинг и т. д.).

В данной статье рассматривается возможность создания недорогой мобильной, расширяемой беспроводной системы отслеживания движений с применением технологии БСС. Использование данной технологии в такой системе является нестандартным применением, ранее она не применялась для подобных задач.

Технология беспроводных сенсорных сетей

Беспроводная сенсорная сеть (БСС) – это беспроводная система, представляющая собой распределенную, самоорганизующуюся и устойчивую к отказу отдельных элементов сеть миниатюрных устройств с автономным источником питания. Узлы такой системы транслируют сообщения друг через друга, обеспечивая значительную площадь покрытия при малой мощности передатчика.

Обмен информацией между узлами системы происходит по беспроводным каналам связи по протоколу ZigBee.

Протокол ZigBee предоставляет беспроводную связь с низким энергопотреблением для множества приложений, которые осуществляют функции наблюдения или управления. Это международный открытый стандарт, контролируемый объединением ZigBee Alliance.

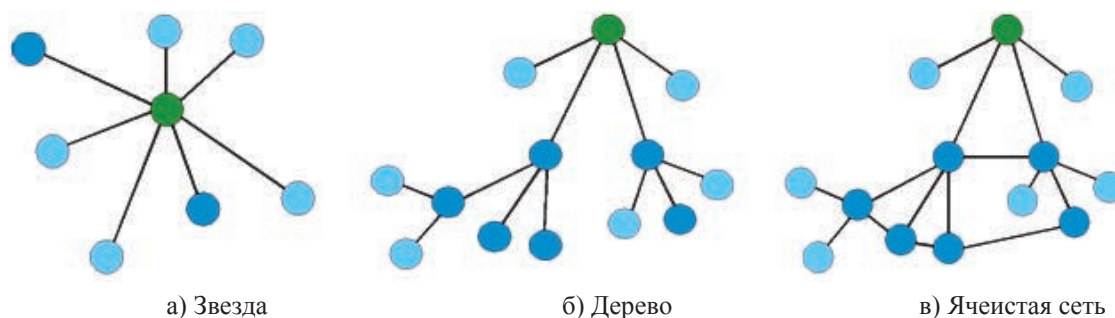


Рис. 1. Поддерживаемые протоколом ZigBee топологии БСС-сетей.

Данный протокол создан на основе стандарта IEEE802.15.4 для пакетной беспроводной передачи данных. Он поддерживает гибкие, расширяемые сетевые топологии (рис. 1); содержит встроенные функции для организации сети и маршрутизации данных; обеспечивает простую установку и высокую устойчивость к сбоям, полноценные меры по безопасности; позволяет преодолеть традиционные ограничения маломощных беспроводных сетевых решений, такие как малая дальность и ограниченное покрытие, а также уязвимость к сбоям в узле и в радиолинии.

Предлагаемое решение

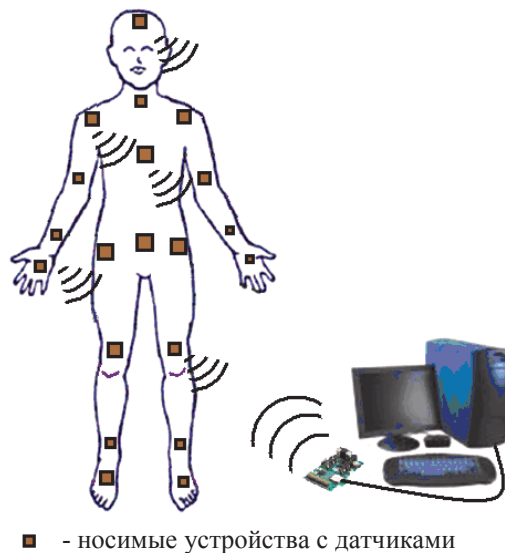
Разрабатываемая система представляет собой аппаратно-программный комплекс (рис. 2). Аппаратная часть системы состоит из трех основных компонент, в том числе:

- 1) комплекта беспроводных носимых конечных устройств с датчиками, закрепляемых на частях тела, движения которых необходимо отследить,
- 2) модуля координатора, который собирает информацию с датчиков конечных устройств и передает ее на сервер обработки и визуализации данных, с которым он соединен, и
- 3) головного компьютера-сервера. Датчики предоставляют информацию об углах поворота, ускорении, скорости, координатах, ориентации, угловой скорости и угловом ускорении. Программное обеспечение системы отвечает за сбор и обработку данных с датчиков движения и визуализацию движений 3-мерной модели человека-аватара в режиме реального времени.

В качестве тестовой системы был разработан макет, позволяющий продемонстрировать передачу данных с датчиков по беспроводному каналу к координатору, который передает их на сервер обработки и визуализации данных. Студия визуализации на основе этих данных визуализирует движения аватара.

Аппаратная часть. Для сбора, первичной обработки и пересылки данных в наших экспериментах используются беспроводные микроконтроллеры английской компании Jennic. Беспроводные ZigBee микроконтроллеры позволяют в реальном времени снимать, первично обрабатывать показания с датчиков и передавать их на сервер обработки и визуализации данных для отображения движений аватара в программной студии 3-мерной визуализации. Макет системы построен

на базе ZigBee микроконтроллера JN-5139 (рис. 3).



■ - носимые устройства с датчиками

Рис. 2. Беспроводная система отслеживания движений человека на основе БСС.



Рис. 3. Отладочная плата микроконтроллера JN-5139 компании Jennic.

Технические характеристики платы микроконтроллера:

Особенности передатчика:

- Частота передачи 2.4 ГГц;
- Шифрование 128-bit AES;
- MAC адресация;
- Интегрированное управление питанием и режимом энергосбережения;
- Напряжение питания 2.2 - 3.6 Вольт;
- Ток потребления в спящем режиме 0.2 мкА;
- Ток потребления в активном режиме 1.3 мкА;
- Чувствительность передатчика - 97dBm;
- Мощность передатчика +3dBm;

Особенности микроконтроллера:

- 32-разрядный RISC-процессор с низким потреблением энергии;
- 192кб ROM (ПЗУ) системной памяти;
- 8кб, 16кб, 32кб или 96кб RAM (ОЗУ) памяти для программного кода;
- 4 12-битовый АЦП, 2 11-битовые ЦАПы.
- 2 UART-а.
- Подключение датчиков по интерфейсам SPI и I2C;
- Рабочая температура $-40^{\circ}\text{C} \div +85^{\circ}\text{C}$;
- Размер модуля 8x8 мм 56-контактный QFN.

В экспериментальной системе в качестве базовых датчиков для измерения движений человека используются миниатюрные MEMS- акселерометры и гироскопы. Датчики, построенные по этой технологии, отличаются малыми размерами, низ-

кой стоимостью, низким энергопотреблением и высокой производительностью.

В качестве датчика ускорения используется интегральный MEMS-акселерометр MMA7455L производства компании Freescale, имеющий следующие характеристики:

- цифровой выход (I²C/SPI);
- малое потребление энергии 400мкА;
- чувствительность выбирается из диапазона: $\pm 2\text{м}/\text{с}^2$, $\pm 4\text{м}/\text{с}^2$, $\pm 8\text{м}/\text{с}^2$;
- корпус размером 3x5x1мм LGA-14;
- низкая стоимость.

В качестве датчика измерения углов поворота используется интегральный MEMS-гироскоп IDG-300 компании InvenSense со следующими характеристиками:

- два независимых гироскопа по осям X и Y;
- полный диапазон измерений $\pm 500^{\circ}/\text{с}$;
- встроенный фильтр нижних частот;
- напряжение питания 3 Вольт;
- корпус размером 6x6x1.4 мм QFN;
- низкая стоимость.

Программная часть нашей системы состоит из двух составляющих: программное обеспечение (ПО) для микроконтроллеров и ПО для визуализации движения на сервере.

ПО для микроконтроллеров было разработано на языке Си, позволяющее снимать показания с датчиков, предварительно их обрабатывать и передавать на сервер обработки и визуализации данных.

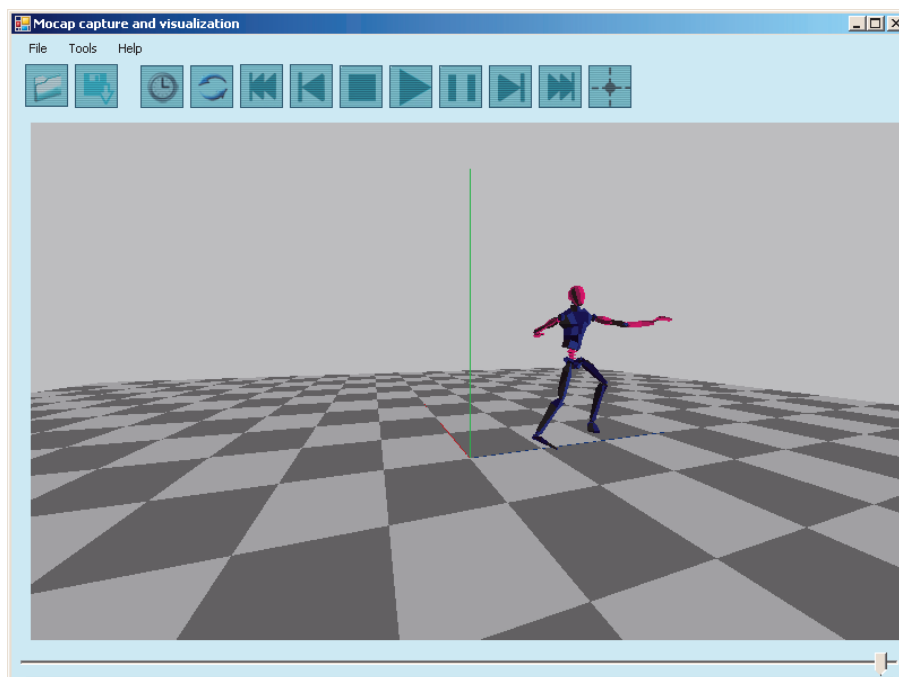


Рис. 4. Студия визуализации отслеживаемых движений человека.

Для визуализации в реальном времени отслеживаемых движений человека была разработана программная студия трехмерной визуализации (рис. 4). Разработка велась на языке MS Visual C# с использованием открытой графической библиотеки OpenGL. Данная студия позволяет в реальном времени визуализировать отслеживаемые движения, сохранять полученные данные и воспроизводить ранее сохраненные данные визуализации (анимации).

Программа совместима с существующими системами трехмерного моделирования, такими как Autodesk 3ds Max, Maya, Blender и др. Это обеспечивается за счет использования для хранения данных файла в формате BVH, разработанного компанией Biovision и широко используемого в системах «захвата движений» типа Motion Capture. Данный формат данных позволяет хранить всю необходимую для 3-мерной визуализации человека-аватара информацию, включая данные о скелете анимируемого персонажа и данные о движениях отдельных частей тела.

3. Особенности предлагаемого решения

Ранее не было полностью беспроводной системы отслеживания движения в реальном времени. В лучшем случае в таких системах носимые конечные устройства с датчиками объединяются проводными каналами, а информация на компьютер передается по беспроводным каналам связи. В данной же разработке базовые элементы системы не связаны проводными каналами связи - все данные с носимых устройств передаются по радиоканалу с частотой 2,4 ГГц. Для использования системы не требуется какого-либо специального помещения с дополнительным оборудованием (специальные камеры, экраны, излучатели, потолочные маркеры и т. п.), достаточно подключить систему к обычному компьютеру.

Среди основных достоинств системы следует отметить относительно низкую стоимость конечной системы по сравнению с существующими подобными системами, отсутствие лишних проводных каналов связи, хорошую масштабируемость, отсутствие потребности в специально оборудованной лаборатории, автономность конечных носимых устройств с датчиками, работу в реальном времени, совместимость с существующими системами трехмерного моделирования.

4. Будущее развитие системы

Спектр возможных применений разрабатываемой системы достаточно широк. К перспектив-

ным областям применения беспроводной системы отслеживания движений человека можно отнести:

- погружающие системы виртуальной реальности, предоставляющие весь комплекс человеко-машинных интерфейсов для взаимодействия человека с виртуальной средой;
- домашние игровые системы, позволяющие в домашних условиях более полно погрузиться в атмосферу игры или «социальной» виртуальной среды типа «Вторая Жизнь» (Second Life);
- профессиональные системы для измерений и обработки данных движения человека в киноиндустрии при создании художественных и анимационных фильмов, в интерактивных телевизионных студиях и в рекламной индустрии;
- тренировочные и симуляционные системы виртуальных сред, позволяющие отрабатывать различные ситуации для задач подготовки специалистов (медицина, подготовка к работе как на борту, так и за его пределами МКС, виртуальные командные центры и т. п.);
- виртуальная робототехника и преобразование реальных движений в виртуальные для управления различными манипуляторами, человеко-подобными роботами;
- исследования в области эргономики, реабилитационной медицины;
- применения в области спорта высших достижений для сбора данных спортсменов в процессе тренировок и соревнований для их последующей обработки и анализа.

Заключение

В данной работе представлена беспроводная система отслеживания движений человека на основе технологии беспроводных сенсорных сетей. Данная система включает в себя все необходимое оборудование и программное обеспечение для отслеживания движения и его визуализации.

Основные достоинства предлагаемого решения – это отсутствие проводных каналов связи, высокая масштабируемость, низкая стоимость по сравнению с существующими системами, а также отсутствие необходимости в оборудовании специальных площадок или помещений.

Разработанный макет системы отслеживания движений человека демонстрирует возможности применения технологии БСС при передачи данных с датчиков по беспроводному каналу связи и визуализации их в реальном времени на сервере визуализации.

В будущем планируется разработать собственный измерительный инерциальный модуль не-

большого размера, который можно было бы закреплять на различных частях тела человека для измерений и преобразования движений человека. В ближайших планах разработчиков переход на более производительную платформу от компании Freescale. Планируется использование системы не только как единого целого, но и использование отдельных компонентов в самостоятельных проектах.

Литература

1. А. Н. Вабищевич. Беспроводная система трехмерной визуализации движения в реальном времени. Тезисы докладов «Научно-технической конференции студентов, аспирантов и молодых специалистов МИЭМ». – М. МИЭМ, 2009г. , с. 145-146.
2. М. М. Комаров, С. Г. Ефремов. Универсальная платформа для беспроводной сенсорной сети. Тезисы докладов «Научно-технической конференции студентов, аспирантов и молодых специалистов МИЭМ». – М. МИЭМ», 2009г. , с. 153-155.
3. Midori Kitagawa and Brian Windsor , MoCap for Artists. Workflow and Techniques for Motion Capture, 2008 Published by Elsevier Inc.
4. M. Meredith, S. Maddock, Motion Capture File Formats Explained, 2001 by Department of Computer Science, University of Sheffield.
5. Технология ZigBee. Ссылка в Internet: <http://www.zigbee.org/en/index.asp>
6. Springer, Guang-Zhong Yang (Ed.) «Body Sensor Networks».
7. Insup Lee, Joseph Y-T. Leung, Sang H. Son, Handbook of Real-Time and Embedded Systems, 2008 by Taylor & Francis Group, LLC.

Опыт использования систем компьютерной графики в тренажерах ПКА

Масалкин А. И. , к. т. н. , доцент

(РГНИИЦПК им. Ю. А. Гагарина, Звездный городок).

В предложенном докладе приводится обзор развития систем компьютерной графики, используемых в качестве средств моделирования визуальной обстановки тренажеров для подготовки космических экипажей в Центре подготовки космонавтов имени Ю. А. Гагарина. Накопленный в течение 20 лет опыт работ в данной области позволил полностью перейти от моделирования визуальной обстановки на оптико-механических принципах к компьютерному моделированию окружающей среды и даже к компьютерной визуализации некоторых режимов работы штатной бортовой аппаратуры пилотируемых космических аппаратов.

Одной из важнейших составных частей тренажеров пилотируемых космических аппаратов, впрочем, как и подавляющего большинства тренажеров других динамических объектов, является система имитации визуальной обстановки (СИВО) в приборах и средствах наблюдения. Обязательными функциями СИВО являются формирование визуальной информации и её воспроизведение в приборе или средстве наблюдения.

Для первых тренажеров пилотируемых космических кораблей создавались уникальные, не имеющие аналогов оптико-механические имитаторы визуальной обстановки, предназначенные для моделирования визуальной обстановки в конкретных средствах наблюдения. В этих имитаторах функции формирования изображения и функции отображения изображения в средстве наблюдения были совмещены, а для контроля изображения использовалась телевизионная камера, которая передавала изображение визуальной обстановки на видеомонитор пульта инструктора. Учитывая уникальность и сложность таких имитаторов, стоимость их была весьма большой, а сроки изготовления растягивались на годы. В 1971 году была запущена первая космическая станция «Салют» и появилась задача подготовки экипажей по осуществлению режимов сближения и стыковки транспортных кораблей со станциями. И в тренажерах в конце 60-х и начале 70-х годов появились первые оптико-телевизионные имитаторы, в которых функции формирования первичного изображения и функции отображения этого изображения в средстве наблюдения были разделены. Прежде всего, это было связано с тем, что

появились штатные бортовые телевизионные системы, в задачи которых, в числе прочих, входил контроль за процессом сближения и стыковкой или с помощью которых непосредственно осуществлялась стыковка.

Объем средств СИВО для тренажеров являлся достаточно большим. Поэтому при разработке средств СИВО требовалось соединить два разноплановых требования: - создание средств для имитации визуальной обстановки в конкретном приборе или средстве наблюдения и, чтобы они, по возможности, были унифицированными, поскольку это является основополагающим принципом централизации средств. Кроме того, технические средства должны были быть достаточно надежными или обеспечены резервированием и должно было быть обеспечено документирование процесса тренировки путем видеозаписи любых нужных сюжетов с помощью видеоманитрофонов.

Наиболее полно требованиям универсальности соответствовали разрабатываемые в ИАиЭ СО АН СССР, так называемые, синтезирующие системы визуализации (ССВ). Это направление развития средств имитации (в частности, система «Горизонт-1» как результат НИР, выполненной в ИАиЭ СО АН СССР) сразу же оценили специалисты, и не случайно уже в 1979 году появился «договор о научно-техническом сотрудничестве между ИАиЭ СО АН СССР, ЦПК и ОКБ СУ НПИ по вопросу создания синтезирующей системы визуализации внешней космической обстановки».

Подключение ИАиЭ СО АН СССР в качестве соисполнителя по созданию синтезирующих систем визуализации в рамках «ТМК» позволило

ЦПК в 1985 году получить первую рабочую ССИВВО «Акса́й». ССИВО «Акса́й» была разработана для задач моделирования визуальной обстановки в передних иллюминаторах тренажеров изделия «Буран». Однако, так как создание этих тренажеров затягивалось, то в 1988 году было решено доработать ССИВО «Акса́й» (визуальные модели) для проведения подготовки операторов по управлению автономным средством перемещения космонавта (АСПК) на тренажере «Дон-21КС». Работа была успешно выполнена и в декабре 1988 года была проведена первая тренировка на тренажере с применением ССИВО «Акса́й». Подготовка операторов по программе пятой основной экспедиции на тренажере «Дон-21КС» прошла хорошо, что и позволило 01 февраля 1990 года летчику-космонавту Сереброву А. А. провести испытания АСПК в открытом космосе. Всего на ССИВО «Акса́й» было проведено около 70 тренировок.

Ограниченные технические возможности ССИВО «Акса́й», низкая надежность не позволяли использовать эту систему ни на тренажерах изделия «Буран», ни на тренажерах транспортного корабля «Союз-ТМ». Поэтому в 1987 году была начата разработка системы компьютерной генерации изображений следующего поколения – синтезирующей системы визуализации «Альбатрос» (ССВ «Альбатрос»). При разработке и изготовлении этой системы (а их изготовлено было только для ЦПК три штуки) был учтен опыт создания «ССИВО «Акса́й». В 1991-1992 ССВ «Альбатрос» были поставлены в ЦПК. Эти системы были предназначены для моделирования визуальной обстановки в средствах наблюдения тренажеров по программе «Буран». Однако, в связи с приостановкой в 1993 году работ по тренажеру изделия «Буран», два комплекта ССВ «Альбатрос» были перенацелены на работу в специализированных тренажерах причаливания и стыковки транспортного корабля «Союз-ТМ». После доработки визуальных моделей с 1 марта 1995 года началась эксплуатация этих систем в составе тренажеров «Телеоператор» и «Дон-Союз-ТМ», и она продолжалась вплоть до 2001 года. При этом формировались изображения объектов стыковки, Земли и звездного неба для оптического визира ВСК-4, бортового телевизионного индикатора КЛ-107 и блистера. С применением систем «Альбатрос» было проведено более 1000 тренировок.

Следует отметить, что при разработке СКГИ «Акса́й» и «Альбатрос» вопросы экономической целесообразности практически не поднимались. Основными требованиями были расширение

функциональных возможностей генераторов по имитации ВВО, увеличение мощности графических процессоров, обеспечение необходимых эксплуатационных характеристик оборудования. Указанные системы представляли собой уникальные разработки, реализованные на специализированных графических процессорах и специально созданном программном обеспечении.

События начала 90-х годов нанесли серьезный удар по работам. Созданная кооперация, за исключением основных соисполнителей в России, практически была разрушена, были прекращены многие перспективные работы, серьезно сокращено финансирование работ. Все это потребовало коренного пересмотра технической политики, а после остановки работ по программе «Буран» стало ясно, что в существующем виде централизованная система ИВО для обеспечения работ по программе «Мир» уже не нужна. Поэтому с середины 90-х годов наступил период небольших доработок, произошло перераспределение функций между средствами СИВО и их приближение к рабочим местам тренажеров. И именно в этот период наиболее ярко обозначились преимущества компьютерных генераторов изображений.

Создавая оптико-телевизионные имитаторы визуальной обстановки и связывая тренажеры телевизионными сигналами вещательного стандарта, специалисты понимали ограниченные возможности этих средств, однако в то время и системы генерации изображения еще не могли составить конкуренцию этим имитаторам вследствие ограниченных возможностей вычислительной техники, низкой надежности, нехватки опыта у разработчиков ССВ. И только третье поколение конкуренцию оптико-телевизионным имитаторам, а заменило их к настоящему времени практически везде.

Таким образом, в конце 90-х годов мы пришли к системам, работающим в составе конкретного тренажера, но созданным с использованием новейших технологий. Вместе с тем, сетевые технологии позволяют, при необходимости, создавать и определенную централизацию средств СИВО (так уже любая из 2-х ССВ «Альбатрос» практически работала на любой из двух тренажеров по любой из двух программ: «Мир» и МКС).

Пятиканальная СКГИ «Ариус-3» была разработана в 1998 году для нового комплексного тренажера ТДК-7СТ № 3 транспортного корабля «Союз-ТМА». В настоящее время после неоднократных доработок она обеспечивает моделирование визуальной обстановки в центральном и

периферийных полях зрения визира специально-го космонавта ВСК-4, на средстве отображения Интегрированного пульта управления (ИНПУ), в иллюминаторах спускаемого аппарата (СА). В 1999 году подготовка экипажей по программе МКС стала проводиться на специализированном тренажере причаливания и стыковки «Дон-Союз-ТМ». Для этого была создана новая СКГИ «Ариус-СТ». После непродолжительной отладки ее в составе тренажера в феврале 2000 года была проведена первая тренировка с применением трех каналов этой системы. В этом же году количество каналов формирования сюжетов визуальной обстановки было доведено до семи, что позволило обеспечить формирование изображений одновременно во всех средствах наблюдения тренажера: в центральном и периферийных полях зрения визира специального космонавта ВСК-4, на средстве отображения ИНПУ, в иллюминаторах СА, блистере бытового отсека (изображения наблюдаемого объекта и фона отдельно). Кроме того, программные средства СКГИ «Ариус-СТ» позволили формировать символическую информацию, генерируемую штатной бортовой системой «Символ-Ц» и отображаемую на средстве отображения ИНПУ.

Мощный импульс для разработки новых СКГИ был дан при создании новых тренажеров российского сегмента МКС. Задачи формирования сюжетов визуальной обстановки в средствах наблюдения тренажера служебного модуля были решены системой «Альтаир». Основное назначение пятиканальной СКГИ «Альтаир» - формирование изображений земной поверхности, так как экипажи проходят подготовку на тренажере и по визуально-инструментальным наблюдениям. В ходе них они должны визуально опознавать географическую местность, наблюдаемую в оптических средствах. Именно поэтому требования к изображению земной поверхности в приборах наблюдения здесь существенно отличаются от требований к другим аналогичным системам. С этой целью в визуальные модели земной поверхности были включены достаточно детальные снимки отдельных природных и техногенных объектов, позволяющие оператору производить многократное (до 10 раз) увеличение рассматриваемого объекта.

Первая система «Ариус-3» явилась промежуточной моделью на пути перехода от специализированных графических устройств к серийным средствам общего назначения. В качестве генератора изображения использовалось оригинальное устройство на базе серийно выпускаемых про-

граммируемых цифровых сигнальных процессоров (DSP). Остальные подсистемы СКГИ были выполнены на серийно выпускаемых устройствах. Помимо создания генератора изображения были решены следующие основные задачи: проектирование общей структуры системы, подбор и сопряжение подсистем, отработка многоканального режима работы, отладка взаимодействия с вычислительной системой тренажера и т. д. Опыт создания СКГИ «Ариус-3» позволил сформировать современные принципы построения СКГИ и отработать технические решения, которые затем были реализованы в последующих системах («Ариус-СТ», «Альтаир», «Антарес», «Арктур»), полностью построенных на серийных устройствах. В качестве геометрического процессора используется стандартный центральный процессор типа Pentium, растеризатора – серийные графические ускорители (акселераторы) типа GeForce. Это позволяет при необходимости наращивать производительность канала путем модульной замены графических подсистем без влияния на остальные составляющие СКГИ, ее общую структуру и взаимодействие с другими системами тренажера.

В настоящее время тренажеры по программе МКС практически полностью обеспечены средствами имитации визуальной обстановки на основе систем компьютерной графики. Указанные средства имеют хороший запас по модернизации относительно несложным способом (заменой центрального процессора и графических акселераторов и соответствующей доработкой программного обеспечения), что потенциально способствует решению всех задач программы МКС. Таким образом, в результате работ по созданию средств СИВО:

были своевременно обеспечены замены устаревших технических средств СИВО новыми системами компьютерной генерации изображений, полностью отвечающим нынешним требованиям, и в этом направлении ЦПК остается лидером; приобретен большой опыт в эксплуатации сложной техники и создания средств СИВО; были решены не только задачи имитации внешней визуальной обстановки, а также и задачи моделирования некоторых бортовых информационных систем с помощью систем компьютерной генерации изображений (СКГИ).

Моделирование движения виртуальных объектов на основе скелетной анимации

Торгашев М. А. , к. ф. м. н. , НИИСИ РАН

(Работа выполнена при поддержке РФФИ, гранты №07-07-00161-а и №06-07-89265-а)

В статье рассмотрена задача моделирования движения сложных иерархически организованных систем с целью имитации их естественного поведения. Для ее решения в качестве базового взят метод скелетной анимации. Для реализации рассмотренной технологии потребовалось реализовать экспорт необходимых параметров системы костей и оболочки объекта из системы трехмерного моделирования, а также метод скелетной деформации в системе визуализации. После осуществления скелетной анимации требуется пересчет нормалей трехмерной модели для обеспечения правильного освещения, поэтому в рамках исследования также был предложен метод пересчета нормалей на основе информации о сглаживании. Исследование и разработка соответствующих методов проводились с упором на достижение максимальной скорости и качества визуализации.

Актуальной задачей исследований в области виртуальной реальности является развитие методов анимации для сложных многообъектных динамических систем. Одной из таких задач является разработка метода качественной анимации сложных иерархически организованных систем с целью моделирования их естественного поведения (для живых персонажей – их ходьбы, бега, жестикуляции). Для решения этой задачи в качестве базового был взят метод скелетной анимации.

Скелетная анимация относится к модификаторам, работающим повершинно. Ее идея состоит в следующем: для деформируемого объекта задается каркас или скелет, представляющий собой систему костей (Bones), чаще всего организованную иерархически. Сам же объект выступает в роли оболочки, «кожи» (Skin), поверх этого скелета. При анимации отдельных костей скелета оболочка повторяет эти движения в той части, где она лежит поверх данных костей, а в местах сочлене-

ний получаем плавное сопряжение поверхностей. Для реализации такой идеологии на этапе моделирования для каждой из вершин оболочки задаются коэффициенты влияния каждой из костей каркаса. Чаще всего при этом есть вершины, полностью относящиеся к одной из костей (соответствующий коэффициент равен 1, а все остальные - 0), в местах же сопряжения костей вершины имеют по несколько ненулевых коэффициентов. Задание коэффициентов костей в системе моделирования чаще всего выполняется не вручную, а с помощью визуальных инструментов, задающих области влияния отдельных костей. Применение скелетной анимации на этапе визуализации для каждой из вершин сводится к вычислению итоговой матрицы деформации как линейной комбинации матриц костей каркаса с соответствующими коэффициентами, и последующему применению этой матрицы к исходной вершине.

Иллюстрация скелетной анимации приведена на рис. 1.



Рис. 1. Скелетная анимация.

Рассмотрим параметры скелетной анимации и алгоритм вычисления для некоторого объекта.

При экспорте параметров объекта в файл сцены записываются следующие параметры:

Параметры экспорта Skin

- Количество костей у объекта-оболочки N ;
- Для каждой из костей $i \in [0, N-1]$, составляющих скелет, записывается: идентификатор i -ой кости (для последующего поиска и привязки при импорте сцены) - ID_i ; исходная матрица узла, соответствующего i -ой кости, на момент применения модификатора Skin (этот параметр используется при расчете скелетной анимации в системе визуализации - M_i^{Init});
- Для каждой вершины объекта - оболочки $i \in [0, N_{verts}-1]$, N_{verts} - число вершин, записываются:

Количество N_i костей, влияющих на i -ую вершину;

Для каждой из этих костей $j \in [0, N_i-1]$ записываются:

- индекс j -ой кости I_{ij} в массиве костей;
- весовой коэффициент j -ой кости w_{ij} ;
- Исходная матрица объекта (перед применением модификатора скелетной анимации) M_{Skin} .

Сцена с экспортированными параметрами далее загружается в систему визуализации для последующего отображения. На этапе загрузки производится привязка костей оболочки к соответствующим узлам сцены с помощью поиска указателя на узел по индексу.

В процессе построения кадра после вычисления матриц узлов и до отправки данных о вершинах на конвейер визуализации производится вычисление скелетной деформации объекта оболочки в конкретный момент времени. Рассмотрим последовательность действий этого процесса:

Процедура CalcSkinning

- Для каждой кости скелета $i \in [0, N-1]$ вычисляется влияющая матрица M'_i на основе текущей модельной матрицы M_i соответствующего узла по следующей формуле: $M'_i = M_{Skin}^{-1} M_i M_i^{Init} M_{Skin}$.
- В цикле по всем вершинам объекта $i \in [0, N_{verts}-1]$ производится вычисление новых координат вершин v'_i на основе

влияющих матриц узлов и коэффициентов соответствующих костей:

Обнуляем результирующие координаты i -ой вершины $v'_i = (0 \ 0 \ 0)$.

В цикле по костям $j \in [0, N_i-1]$, влияющим на i -ую вершину, пересчитываем координаты координаты по формуле:

$$v'_i = \sum_{j=0}^{N_i} w_{ij} M'_{I_{ij}} v_i$$

После выполнения всех модификаторов, работающих над вершинами, производится перерасчет нормалей объекта, что обеспечивает правильность освещения деформированного объекта.

Нормаль к конкретному треугольнику находится с помощью векторного произведения векторов ребер (см. рисунок 2), выбранных таким образом, чтобы результирующий вектор был ориентирован в направлении лицевой стороны. Поскольку алгоритмы вычисления освещенности требуют нормированных величин нормалей, результат приводится к единичной длине.

Таким образом, алгоритм вычисления нормали отдельного полигона имеет следующий вид.

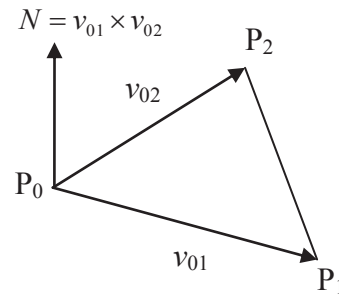


Рис. 2. Вычисление нормали треугольника.

Алгоритм вычисления нормали полигона

- Вычисляем векторы $v_{01} = (P_1 - P_0)$ и $v_{02} = (P_2 - P_0)$
- Вычисляем их векторное произведение: $N = v_{01} \times v_{02}$
- Нормализация результата: $\|N\| = \sqrt{N_x^2 + N_y^2 + N_z^2}$ $N = N/\|N\|$

Рассмотрим алгоритм пересчета нормалей для всего объекта - оболочки с учетом возможного сглаживания ребер полигональной модели.

Алгоритм пересчета нормалей

- Обнуляем координаты всех нормалей объекта $i \in [0, N_{norms} - 1]$, где N_{norms} - количество нормалей: $N_i = (0 \ 0 \ 0)$.
- В цикле по треугольникам объекта $i \in [0, N_{faces} - 1]$, где N_{faces} - количество треугольников:

Вычисляем нормаль к i -ой грани N_i по рассмотренному выше алгоритму.

Если грань - без сглаживания, то записываем в соответствующие нормали вершин нормаль граней: $N_{v0} = N_{v1} = N_{v2} = N_i$;

Иначе - цикл по вершинам треугольника j :

- Добавляем к соответствующей нормали вершины нормаль к текущей грани:

$$N_{vj} = N_{vj} + N_i.$$

- Приведение всех нормалей объекта $i \in [0, N_{norms} - 1]$ к единичной длине $N_i = N_i / \|N_i\|$.

Описанный алгоритм был успешно реализован в системе визуализации [1]. Дополнительно к возможностям анимации, задаваемых системой моделирования, также предложены новые возможности по регулированию воспроизведения анимационных треков в системе визуализации. Анимация отдельных параметров может быть запущена в необходимое время и с нужной скоростью, что позволяет реализовать сложный сценарий поведения сцены в моделируемой динамической обстановке. Для реализации этой возможности в системе визуализации введено понятие именованного анимационного трека, к которому привязываются анимация одного или нескольких параметров. У каждого из таких треков вводится свое локальное время анимации, не зависящее от глобального времени, относящегося к остальным параметрам сцены. Управление локальным временем трека, изменение коэффициента скорости воспроизведения, его запуск и остановка выполняются с помощью команд на встроенном скриптовом языке. Эти команды могут поступать как результат действий пользователя (ручное задание команд), так и от отдельных модулей системы (например, от модуля сетевого взаимодействия, принявшего внешнюю команду; либо, например, от модуля динамики). Такие возможности обеспечивают максимальную гибкость при использовании анимации и позволяют реализовать произвольные сценарии анимации сложных иерархически организованных систем с целью моделирования их естественного поведения.

Иллюстрация работы предложенных методов в системе визуализации [1] для моделирования движения человека представлена на рис. 3.

Литература

1. Торгашев М. А., Михайлюк М. В., Фам Кань Зыонг (Pham Canh Duong), Фам Хонг Куанг

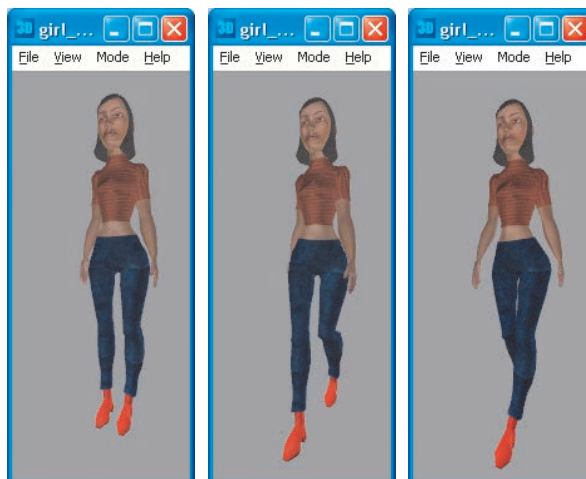


Рис. 3. Моделирование движения человека.

- (Pham Hong Quang). Система трехмерной визуализации виртуальных сцен GLView. Сборник трудов НИИСИ РАН, 2007.
2. Никифоров В. М., Торгашев М. А. Реализация контроллера следования по пути из системы 3D Studio MAX. // Информационные технологии и вычислительные системы, №1, 2007, стр. 20-29.

Особенности управления робототехническими системами космического назначения

Богомолов В. П., к. т. н., ФГУП ЦНИИМАШ

Космическая робототехника – это область научно-практической деятельности по созданию и эксплуатации робототехнических систем космического назначения, предназначенных для обслуживания и сборки космических объектов, проведения испытаний космической техники, автоматизации производственно-технологических процессов в условиях космического пространства. Космическая робототехника обеспечивает повышение безопасности деятельности экипажей космических кораблей, позволяет сокращать сроки создания и отработки космической техники, увеличить сроки активного существования космических объектов, существенно расширяет их функциональные возможности в процессе эксплуатации.

Повышение эффективности обслуживания КО достигается за счёт увеличения точности, скорости и надёжности выполняемых робототехническими системами операций, внедрения технологий дистанционного и автоматического управления, частичного или полного снятия ограничений на длительность операций, увеличения объёмов

выполняемых работ и повышения степени их автоматизации. Указанные направления позволят сократить объёмы рутинных работ, выполняемых экипажами КО при внекорабельной деятельности (ВКД) и внутри гермоотсеков, повысить безопасность выполняемых операций, расширить перечень робототехнических операций, обеспечивающих обнаружение и ликвидацию нештатных и аварийных ситуаций, а также операций по обслуживанию космонавтов при работах как в открытом космосе, так и внутри гермоотсеков.

Использование робототехнических систем (РТС) и комплексов освободит экипажи пилотируемых объектов от выполнения рискованных, трудоёмких и рутинных работ, увеличивая тем самым время, отводимое для решения творческих, научных и прикладных задач, проведения исследований и экспериментов в космосе.

Управление РТС первого поколения осуществлялось по “жёстким” программам, позволяющим выполнять, как правило, только узкоспециализированные операции.

Таблица 1. Условия функционирования объектов в космическом пространстве и возможности робототехнических средств космического назначения

Условия функционирования объектов в космическом пространстве	Возможности робототехнических средств космического назначения
Экстремальные внешние условия, обусловленные воздействием факторов космического пространства	Работа в широком диапазоне внешних условий, в том числе опасных или недопустимых для человека
Необходимость привлечения значительных энергетических, материальных и информационно-телекоммуникационных ресурсов для обеспечения пилотируемых миссий	Экономия материальных и энергетических ресурсов в связи с отсутствием систем жизнеобеспечения
Экстремальные психофизиологические условия работы космонавтов	Замещение человека при выполнении рутинных операций
Длительные сроки транспортировки на сверхбольшие расстояния и пребывания на космических объектах	Возможность перехода в режим пониженного потребления энергии и других ресурсов

Оснащение РТС второго поколения интеллектуальными сенсорными системами позволило существенно расширить количество и номенклатуру операций, выполняемых РТС, гибко перестраивать программу управления, в том числе по командам человека-оператора. На рубеже XX и XXI столетий начались проектно-поисковые разработки интеллектуализированных РТС третьего поколения с постепенной передачей функции оперативного управления от человека-оператора в систему автоматического управления. Конечной целью этих работ является создание автономных интеллектуальных РТС с интерактивным интерфейсом, обеспечивающим дистанционное взаимодействие робототехнических средств с человеком-оператором.

Анализ условий функционирования объектов в космическом пространстве и возможностей робототехнических средств космического назначения показывает обоснованность и перспективность использования современных отечественных и зарубежных достижений робототехники в космической деятельности (см. *Таблица 1*).

Основными целями создания РТС космического назначения (РТС КН) являются:

- повышение эффективности эксплуатации пилотируемых и автоматических космических средств с расширением их функциональных возможностей;
- обеспечение безопасности экипажей орбитальных и напланетных комплексов;

- создание технологий выполнения операций в экстремальных условиях космического пространства для осуществления космической деятельности.

Анализ особенностей применения РТС в экстремальных условиях космического пространства позволяет сформулировать следующие основные принципы управления РТС КН человеком оператором (рис. 1).

Реализация данных принципов возможна только при широком внедрении в процесс управления информационно-коммуникационных технологий с системами дистанционного телеприсутствия и технологий виртуальной реальности.

Информационно-коммуникационные технологии позволяют осуществлять дистанционное управление роботами, в то время как использование технологии виртуальной реальности охватывает все этапы жизненного цикла робототехнических систем от этапа проектирования до изготовления, испытания, эксплуатации и утилизации.

На этапе проектирования после создания графических моделей и моделей описывающих функционирование РТС в заданных условиях космического пространства (невесомость, вакуум, гравитационные воздействия, параметры атмосферы и радиационной обстановки и т. д.) возможно предварительное испытание всех его систем. Наличие сценария, описывающего выполнение миссии позволяет осуществлять моделирова-

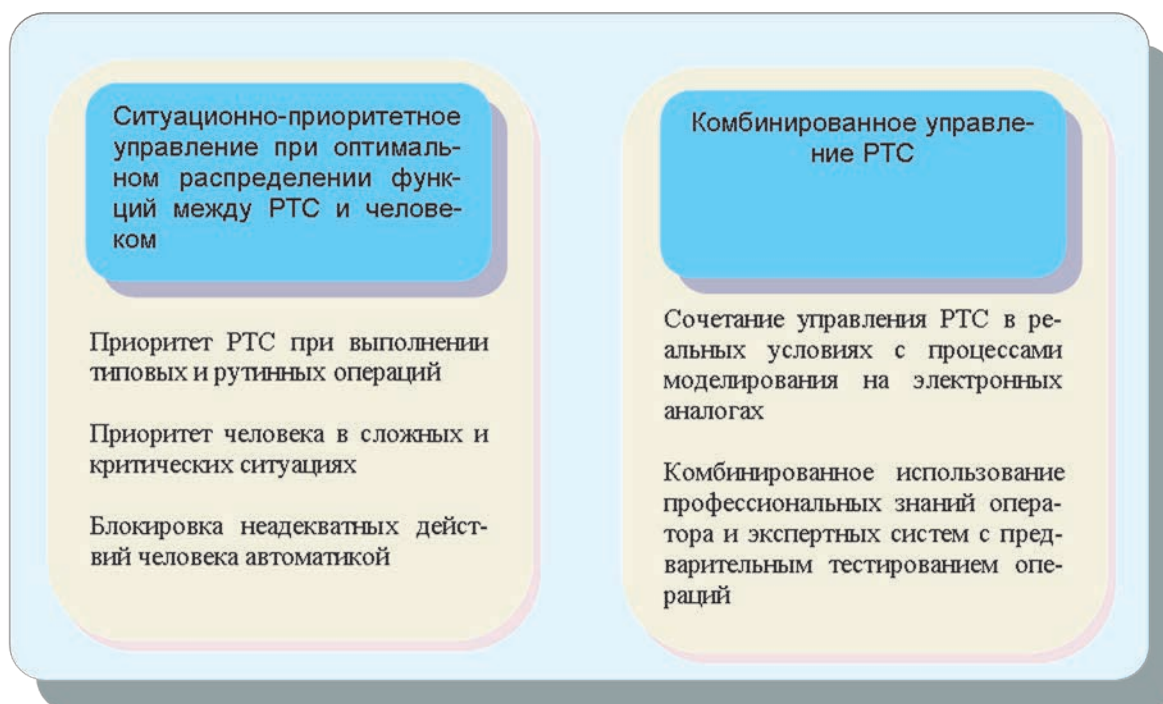


Рис. 1. Принципы управления РТС КН человеком-оператором

ние процессов выполнения конкретных операций в штатном и аварийных режимах работы.

Проведение таких испытаний с натурными макетами в наземных условиях часто невозможно или требует создания сложных дорогостоящих стендов и тренажёров.

В последнее время интенсивно развиваются технологии виртуальной реальности (VR) и телеприсутствия. Системы телеприсутствия обеспечивают передачу телевизионного и стереотелевизионного изображения на большие расстояния. Системы виртуальной реальности обеспечивают решение задачи визуализации операций, выполняемых техническими устройствами путём создания геометрических моделей поверхностей объектов, образующих 3D-сцену и создание программно-технических средств синтеза изображений. Обе технологии позволяют создавать современные тренажные системы и системы обучения, уменьшающие риск ошибочных действий оператора, управляющего техническим устройством.

Вместе с тем использование технологий виртуальной реальности потребует решения целого ряда новых задач, таких как создание:

- прогнозирующих дисплеев для моделирования задержек в линиях связи,
- развитого «человеко-машинного» интерфейса для управления роботами с помощью многомерных указателей, речевых команд, органов управления с силовой моментным и тактильным чувствлением, систем стереовизуализации наблюдаемых процессов, виртуальных аналогов сенсорных систем, позволяющих контролировать поведение робота и состояние его внешней среды.

Важно отметить, что технологии виртуальной реальности позволяют визуализировать невидимые процессы, что существенно расширяет сферу использования РТС КН, упрощают процесс управления за счёт оперативной привязки виртуальных операций к реальным событиям, а также обеспечивают более полное представление о выполняемых операциях.

Разработка интерактивных виртуальных сред (ИВР) моделирования и поддержки функционирования робототехнических средств для сборки и обслуживания космических объектов, включая задачи обслуживания экипажей космических станций, необходима для создания комплекса средств интерактивного моделирования и поддержки процессов дистанционного обслуживания и сборки КО, а также обслуживания экипажей РС МКС с помощью робототехнических средств.

Существует большая группа задач, связанных с обслуживанием пилотируемых комплексов с помощью робототехнических систем, в частности:

- Работы по инспекции состояния внешней поверхности РС МКС, контролю электромагнитной и радиационной обстановки на станции и в её окрестности, проведение экспериментов на станции без участия или с минимальным участием экипажа, обслуживание экипажа как внутри станции, так и при ВКД на орбитальных станциях и на поверхности Луны и Марса.

Появление роботов в системе обслуживания не означает, что все работы будут выполняться автоматически. Система остаётся «человеко-машинной», так как в контуре управления РТС остаётся оператор, у которого изменяется вид деятельности. Избавляясь от выполнения рутинных и опасных для здоровья операции космонавты должны владеть технологиями VR для дистанционного управления сложными робототехническими комплексами.

При подготовке космонавтов к новому для них виду деятельности необходимо создание различных тренажёров, функционально-моделирующих стендов, систем визуального отображения окружающей среды, в которой планируется функционирование РТС, в том числе при совместной работе с космонавтами.

Использование интерактивных виртуальных сред (ИВС), обеспечивает погружение пользователя в трёхмерную интерактивную среду изучаемого явления или процесса, участие в событиях среды с использованием естественного интуитивного интерфейса для взаимодействия с объектами (искусственными или реальными) в виртуальной среде.

Реализация ИВС позволит поддерживать непосредственный контакт с моделируемой средой, обеспечивая полисенсорное восприятие среды, отрабатывать совместную работу экипажей и операторов центров дистанционного управления робототехническими устройствами космического назначения.

Исследования и разработки в данной области носят междисциплинарный характер и наиболее интенсивно ведутся в США, Канаде, Европейском сообществе, Японии, при этом разрабатываются как компоненты систем виртуальной реальности (системы формирования объёмных изображений, системы имитации взаимодействия с объектами виртуальной реальности), так и комплексные тре-

нажеры и модели систем (например, авиационные тренажеры фирмы BOEING, космические тренажеры NASA, NAVY simulator и т. п.) Данными проблемами интенсивно занимаются также фирмы Virtual Reality, Virtual Research, Kaiser Electro-Opt, Raytheon Company (США), Planar, Thomson Training & Simulation, «ТЕХНО», «ЛОГОС», ПКБМ (г. Пенза) и др. Сегодня в мире действует более пятисот полномасштабных установок интерактивного виртуального окружения, которые реально приносят ощутимую пользу в своих областях применения. Наиболее серьезные результаты получены на космических тренажерах, в системах подготовки экипажей атомных подводных лодок и военных кораблей, при обучении операторов атомных электростанций, при исследованиях и разработке нано-технологий.

При использовании же технологии виртуальных сред разработчик изделий значительно лучше понимает и «чувствует» конструкцию, когда он может проникнуть в пространство модели и там непосредственно манипулировать её параметрами.

Для функционирования распределенной человеко-машинной системы, в состав которой входят несколько экспертных групп, участвующих в принятии решений по управлению РТС целесо-

образно создание у различных пользователей виртуальных копий реальных объектов. При этом в адрес пользователя оперативно доставлять только вектор состояния виртуальной среды и реальных объектов. По информации, содержащейся в векторе состояния во всех узлах сети синхронно независимо от других пользователей выполняется перевод виртуального объекта в состояние воспроизводящее состояние реального объекта. Такая схема взаимодействия различных пользователей позволяет синхронно осуществлять анализ работы служебных и целевых систем робототехнического комплекса.

Вектор состояния реального объекта формируется только в центральном серверном узле и является общим для различных пользователей, в узлах которых на основе вектора состояния оперативно формируется виртуальная среда.

Создание комбинированных систем управления РТС КН с широким использованием технологии виртуальной реальности позволит коренным образом улучшить информационное обеспечение операторов РТС, находящихся как в наземных центрах управления, так и на борту орбитальных или напланетных комплексов.

Моделирование и визуализация взаимодействия 3D-объектов сложной структуры в индуцированной виртуальной среде

Афанасьев В. О., *д.ф.-м.н., с.н.с., ЦУП (ЦНИИМаш), МФТИ*

Клименко С. В., *д.ф.-м.н., профессор, МФТИ*

Фомин С. К., *аспирант, МФТИ*

Рассматривается подход к проектированию программной реализации алгоритмов моделирования взаимодействия виртуальных 3D-объектов с объединениями и распадами их структур и изменениями отношения порядка в этих структурах. Предложен подход к построению паттернов проектирования программных систем для управления поведением компонент связности графа сцены на основе возможностей RTR-деревьев. Работа поддержана грантом РФФИ 08-07-00220.

Возможности высокореалистичной визуализации в системах виртуальной реальности существенно расширяют возможности практического применения этих систем на практике. К числу таких возможностей можно отнести наблюдение за реальными объектами на основе данных 3D-позиционирования объектов, получаемых любыми доступными способами (телеметрия, системы GPS/ГЛОНАСС, сенсорные сети и др.). В этом случае объектом наблюдения фактически является не сам объект, а его виртуальная копия – 3D-модель, поведение которой с достаточной для решения целевой задачи точностью воспроизводится (индуцируется) в виртуальной среде, [1] – [3].

До настоящего времени при моделирование поведения 3D-объектов недостаточное внимание уделялось таким аспектам поведения реальных объектов, как их контактное взаимодействие, в результате которого может существенно меняться не только внешний вид объектов, но и модель их поведения. При переносе модели поведения объектов в виртуальную среду, может сильно изменяться семантика данных, на основе которых должно выполняться моделирование. Особенности моделирования и 3D-визуализации поведения становятся особенно важными, если речь идёт о наблюдении за объектами, имеющими сложную динамическую структуру (такими как манипуляторы, андрониды и т.п.), которая может сильно изменяться в ходе моделирования.

Характерным примером процесса взаимодействия объектов, сопровождающегося изменением их структуры, является процесс стыковки модулей ФГБ «Заря» и Node «Unity» Международной космической станции (МКС), выполнявшийся посредством манипулятора, установленного на орбитальном корабле Space Shuttle.

На рис.1 показаны кадры с изображениями виртуальной среды в процессе стыковки. На рис. 2 показано как изменяется граф сцены в ходе моделирования поведения виртуальных объектов. Здесь следует обратить внимание на изменения компонент связности сцены, а также на изменение направлений связей и ролевых функций («ведущий» и «ведомый») моделируемых объектов.

Особенностью процесса стыковки является изменение активности модулей FGB и Node. Перед стыковкой модуль FGB мог перемещаться и маневрировать автономно, модуль Node был закреплён на стыковочном узле орбитального корабля Space Shuttle (рис. 1а и рис.2а). В процессе стыковки модуль FGB сначала присоединился к манипулятору орбитального корабля, затем был соединён с модулем Node. Модули FGB и Node образовали вместе с орбитальным кораблём Space Shuttle единую связку (фрагменты (б) и (в) на рис.1 и 2). После завершения стыковки модуль FGB стал активным, модуль Node присоединился к FGB, а Space Shuttle стал двигаться автономно (рис. 1г, рис. 2г).

В виртуальной среде изменения ролевых функций проявляются в виде изменений источников данных о координатах 3D-объектов. Источником данных для модели активного (ведущего) объекта является модель движения этого объекта

в 3D-пространстве сцены. Для пассивного (ведомого) объекта источником данных является ведущий объект, к которому ведомый объект приклепляется.

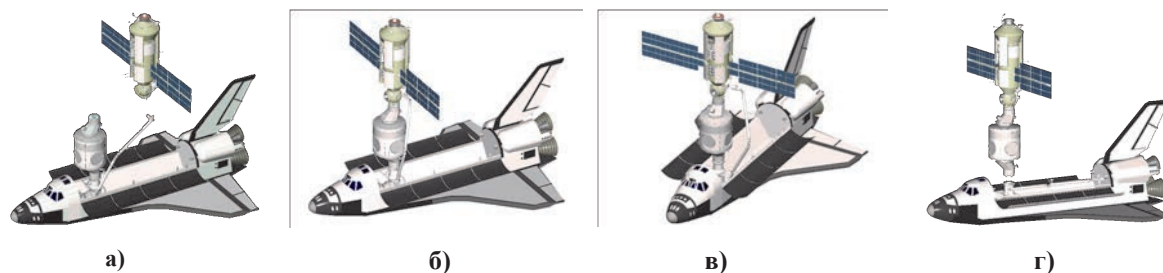


Рис. 1. 3D-сцена моделирования стыковки модулей FGB, NODE и корабля Space Shuttle.

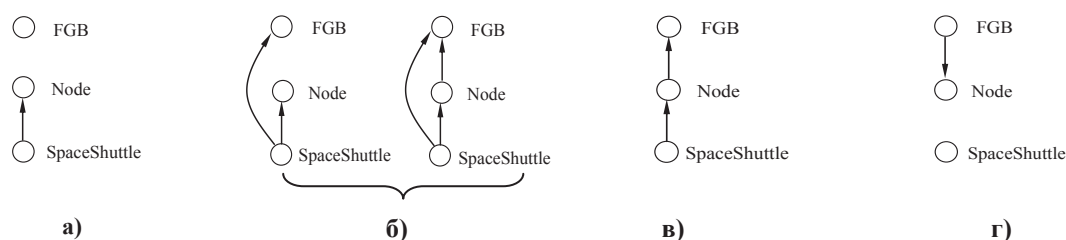


Рис. 2. Изменение компонент связности графа 3D-сцены показанной на рис.1.

Задача моделирования и визуализации поведения взаимодействующих объектов имеет несколько аспектов. Наиболее важными являются следующие:

- необходимо слежение за образованием и разрывом связей между объектами;
- необходимо управлять подключением и отключением источников данных о движении моделей при изменении компонент связности;
- необходимо учитывать возможные изменения отношения порядка в структурах объектов.

Первые два аспекта (изменение компонент связности и источников данных о движении) в той или иной мере уже нашли своё решение и не связаны какими-либо принципиальными проблемами (GOF-паттерны управления объектами или их модификации и т.п. [4, 5]). Третий аспект (изменение отношения порядка в структуре) впервые рассмотрен в [6] и [7], где для этого были введены понятия и определения RTR-деревьев и связанных RTR-списков.

На рис. 3 приведены типичные примеры изменения структуры, которые могут происходить при взаимодействии 3D-объектов, и сопровождаться изменением отношения порядка (фрагменты (в), (г)). Заметим, что такие изменения возможны и при сохранении связности (фрагмент (д)).

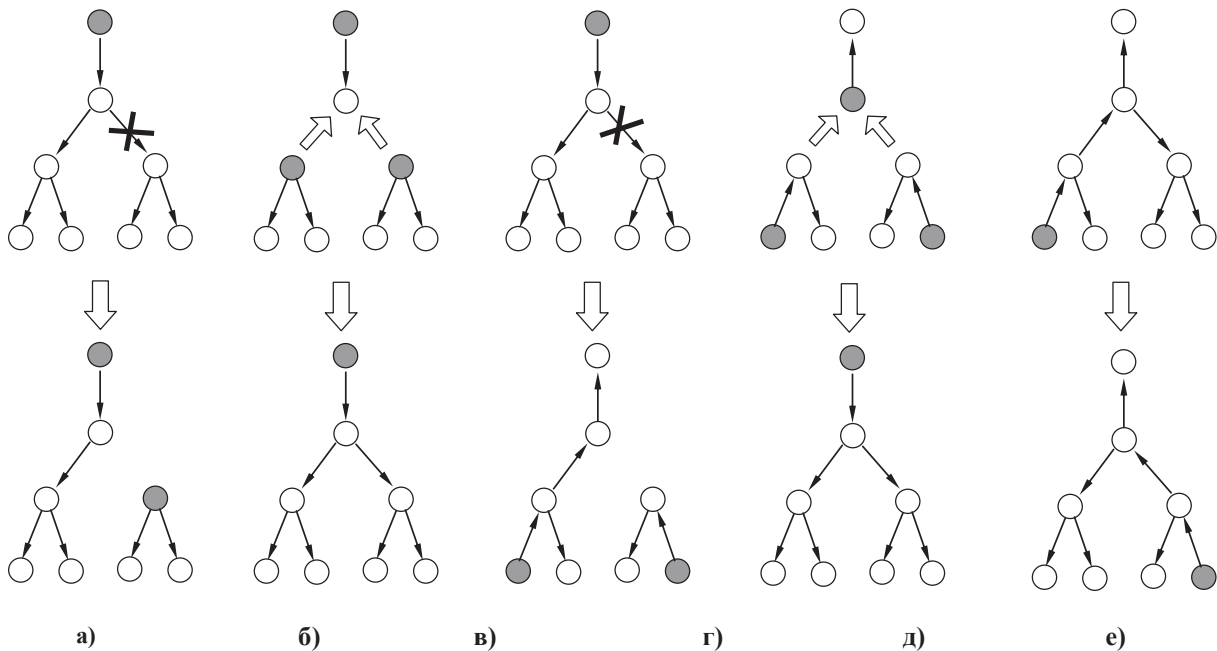


Рис. 3. Примеры типичных изменений компонент связности графов структуры (серым обозначены корни ордеревьев).

Если структура 3D-объекта (вся целиком или её подмножество) используется как кинематическая система (например, при моделировании работы шагающих манипуляторов, [6], [7]), отношение порядка в структуре объекта начинает играть важную роль. Это отношение, в сущности, определяет весь алгоритм управления связанным движением звеньев кинематической системы.

Напомним, что ордеревья структуры 3D-объекта являются нагруженным, причём нагрузкой являются матрицы ортогональных преобразований систем координат структурных единиц поверхности 3D-объекта. Обход ордеревья (в направлении от корня к листьям) является одновременно и процессом рекурсивного вычисления матриц ортогональных преобразований для каждого из примитивов, образующих поверхность. Изменение отношения порядка в таком графе приводит к изменению одновременно и алгоритма обхода, и алгоритма вычисления матриц ортогональных преобразований (ортогональные преобразования на RTR-деревьях подробно рассмотрены в [6] и [7]).

При программной реализации алгоритмов управления компонентами связности графа сцены важной задачей является построения паттернов проектирования, описывающих свойства и взаимодействие программных объектов. Готового паттерна для этой цели в настоящее время не существует. Ближе всего к особенностям рассматриваемой

задачи находятся GOF-паттерны поведения, известные под названиями «Mediator» и «Observer», [4], однако они не обладают исчерпывающим набором возможностей. Для управления взаимодействием авторы предлагают два специальных паттерна, которые названы «Сцепщик RTR-деревьев» (англ. вариант - «RTR-Coupler») и «Расщепитель RTR-деревьев» (англ. вариант - «RTR-Splitter»).

В основу обоих паттернов, положена идея сохранения инкапсуляции алгоритмов рекурсивных ортогональных преобразований в том же виде, что и при обходе монолитного ордеревья, но для структур, связность которых может изменяться в течении сессии моделирования группового поведения 3D-объектов (примерно так, как показано на рис. 3). Для реализации этой идеи имеется предпосылка: исчерпывающая информация о связности ордеревья заключена в списке рёбер, а сам RTR-список может быть виртуальным (заметьте, что списки рёбер не являются связанными списками, см. [6] и [7]).

Таким образом, изменение компонент связности леса 3D-сцены можно реализовать:

- добавлением или исключением мостов в списках рёбер для ордеревьев 3D-объектов;
- объединением и разбиением списков рёбер (заметьте, что списки рёбер не являются связанными списками, см. [6] и [7]).

Для выполнения этих функций целесообразно использовать объект, контролирующий связи внутри RTR-деревьев, описывающих структуры 3D-объектов, а также объект, контролирующий связи 3D-объектов с источниками данных о движении.

Организация паттернов RTR-Coupler и RTR-Splitter проиллюстрирована UML-диаграммами на рис. 4 и рис.5, соответственно. Показаны связи между объектами и функциональная нагрузка этих связей (для обозначения объектов и функций использованы интуитивно-понятные названия).

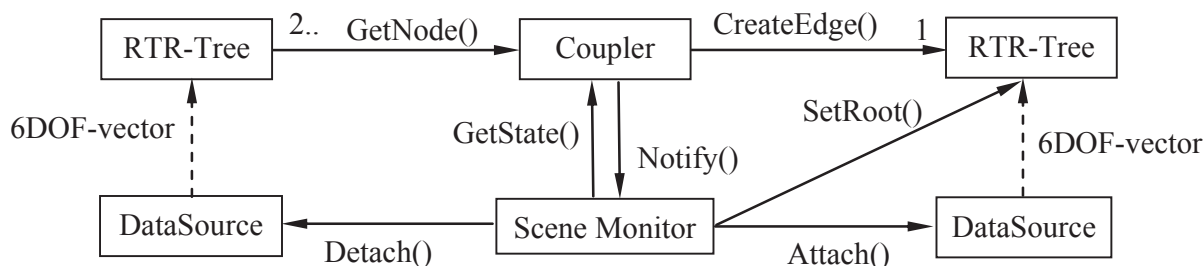


Рис. 4. Схема организации и функционирования паттерна RTR-Coupler.

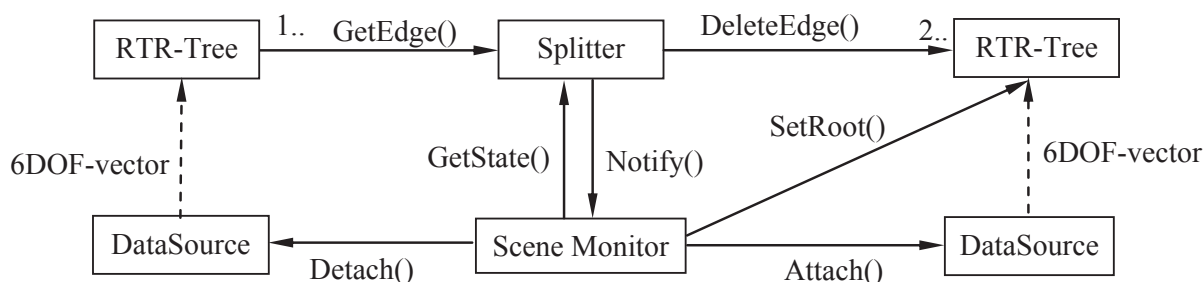


Рис. 5. Схема организации и функционирования паттерна RTR-Splitter.

Необходимо отметить, что в результате работы объектов Coupler и Splitter не уничтожаются существующие и не создаются новые 3D-объекты. Создаются и уничтожаются только мосты между узлами ордеревьев (функции CreateEdge(), DeleteEdge()) и источниками данных о движении (функции Attach(), Detach()). В индуцированной виртуальной среде активные (ведущие) объекты получают данные о движении в виде 6DOF-векторов или полученных на их основе матриц от объектов DataSource, связанных с источниками телеметрической информацией. Пассивные (ведомые) объекты получают эти данные как поддеревья (другими словами, передача данных инкапсулируется в процедурах обхода ордеревьев).

Управление связями между 3D-объектами и источниками данных о движении возлагается на объект SceneMonitor, который также переназначает при необходимости корни в реорганизованных структурах 3D-объектов (функция

SetRoot()). В результате определения корней автоматически задаются нужные направления обходов в ветвях ордеревьев полученных структур.

Между объектами Splitter/Coupler и SceneMonitor организована двусторонняя связь. Соответствующий преобразователь связей получает от монитора сцены данные о состоянии сцены, необходимые для изменения структур (функция GetState()). В свою очередь монитор сцены получает уведомления об изменениях в графе сцены (функция Notify()), необходимые для переключения источников данных о движении между нужными источниками и приёмниками.

Свойства и возможности RTR-деревьев и их программного аналога – связанных RTR-списков – позволяют существенно облегчить моделирование и визуализацию объектов, в структурах которых происходят изменения связности и отношения порядка. В то же время представление об отношении порядка в ордереве как о переменной

величине требует расширения семантики операций над ордеревьями, изменяющими их связность, - в частности операций склейки и разбиения. Задача алгоритмической и программной поддержки этих операций пока не рассматривалась, и в данной работе сделан шаг в этом направлении. Основным результатом использования паттернов RTR-Coupler и RTR-Splitter на практике стало существенное уменьшение трудоёмкости проектирования, разработки и отладки программного кода для моделирования взаимодействия 3D-объектов на основе свойств и возможностей RTR-деревьев и связанных RTR-списков.

Литература

1. Алешин В.П., Афанасьев В.О. Байгозин Д.А., Батурич Ю.М. и др. Система визуализации индуцированного виртуального окружения для задач исследования космоса: состояние проекта//В сб. Трудов 14-й Международной конференции по компьютерной графике и зрению «Графикон-2004», М.: Изд-во МГУ, 2004, 318с., с.12-15.
2. Афанасьев В.О. Модели структур и данных для решения задач управления поведением и взаимодействием объектов в индуцированной виртуальной среде. - Космонавтика и ракетостроение, №2(39), Изд-во ЦНИИМаш, 2005. с.168-179.
3. Afanasiev V. O., Baygozin D. A., Baturin Y. M., Burlakov S. K., Klimenko S. V. Distributed Systems Software Architecture for Induced Virtual Environments. International Conference of Cyberworlds. CW 2005, 23-35 November 2005, Singapore, IEEE Computer Society Press, pp. 378-382.
4. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного программирования. Паттерны проектирования.- СПб.: Питер, 2004.- 366с.
5. Шаллоуей А., Тротт Дж. Шаблоны проектирования. Новый подход к объектно-ориентированному анализу и проектированию.: Пер. с англ.-М.: Издательский дом «Вильямс», 2002.- 288с.
6. Афанасьев В.О. Деревья и связанные списки с переменным отношением порядка (RTR-структуры). - Программирование, т.32, № 6, 2006.
7. Afanasiev V., Baigozin D., Kazanski I., Fomin S., Klimenko S. RTR-Trees for Space Robotics Behavior Simulation and Visualization. VISUAL COMPUTER 2006. Special Issue, 2006, IEEE Computer Society Press.

Пространственно-временное моделирование и планирование индустриальных проектов

Семенов В. А., Аничкин А. С., Казаков К. А., Тарлапан О. А.

Институт системного программирования РАН, e-mail: step@ispras.ru

Работа посвящена проблемам многофакторного анализа и моделирования масштабных индустриальных проектов. В отличие от традиционных задач календарно-сетевое планирование обсуждаемые постановки учитывают пространственно-временной фактор в организации работ и позволяют более адекватно воспроизвести картину будущего проектного решения. В работе рассматривается несколько содержательных постановок задач 4D анализа, моделирования и планирования, решение которых приводит к качественному улучшению процессов управления проектами.

Реализация крупных проектов, таких как возведение небоскреба, строительство атомной электростанции или сборка авиалайнера, невозможна без адекватного планирования, эффективного контроля и управления. Для решения подобных задач традиционно применяются методы календарно- сетевого планирования, такие как метод расчета критического пути (Critical Path Method) и метод оценки и анализа рисков программ (Program Evaluation and Review Technique) [1, 2]. В основе методов лежит идея определения проектных работ и их последовательностей, имеющих минимальный временной резерв и являющихся критичными по отношению к срокам реализации проекта. С этой целью в СРМ методе последовательно уточняются допустимые даты начала и завершения работ, а также устанавливаются временные резервы, превышение которых может привести к сдвигу сроков всего проекта. В PERT методе анализируются риски несвоевременного завершения проекта с учетом вероятностного распределения длительностей отдельных работ. В некоторых модификациях методов принимается во внимание фактор конкурентного использования проектных ресурсов в работах, проводимых одновременно. В ряде случаев это приводит к коррекции календарно- сетевого графика и удлинению сроков проекта.

Для визуального представления результатов планирования обычно применяются диаграммы Ганта, сетевые диаграммы и графики потребления ресурсов. На диаграммах Ганта, являющихся наиболее распространенной визуальной нотацией для проектных планов, работы представляются горизонтальными полосками, позиционируемыми

вдоль временной оси в соответствии с календарными датами начала и завершения. Отношения предшествования между работами изображаются соответствующими соединительными линиями. В альтернативном представлении сетевой диаграммы отдельные работы отображаются в виде именованных вершин графа, а отношения предшествования между ними — в виде направленных ребер. Графики потребления показывают, каким образом назначенные ресурсы распределяются между работами и расходуются в ходе выполнения проекта.

Упомянутые методы планирования и визуализации планов программно реализованы и включены в состав практически всех современных систем управления проектами, таких как MS Project, Primavera, Time Line, Open Plan, Asta PowerProject, Spider Project. Как правило, компьютерные реализации учитывают иерархическую структуризацию и декомпозицию проектных работ, отношения предшествования между ними, временные ограничения в виде установленных промежуточных вех и сроков завершения проекта, назначения и профили потребления ресурсов, а также функции распределения длительности работ. Развернутый сравнительный обзор систем данного класса можно найти в [3].

Вместе с тем, все рассмотренные выше методы игнорируют пространственно-временные ограничения в размещении и использовании ресурсов и не позволяют адекватно спланировать проекты, в которых подобный фактор может иметь решающее значение. Чтобы решить эту проблему, предлагается консолидировать данные календарно- сетевого планирования и конструкторской доку-

ментации, подготовленной в системах автоматизированного проектирования САПР. В результате ассоциирования проектных ресурсов и работ с соответствующими геометрическими моделями формируется единый согласованный пространственно-временной план проекта, который может быть визуализирован, проанализирован, верифи-

цирован, документирован в виде иллюстрационных материалов, серий изображений, видеопрезентаций и при необходимости проконтролирован непосредственно на проектной площадке [4, 5].

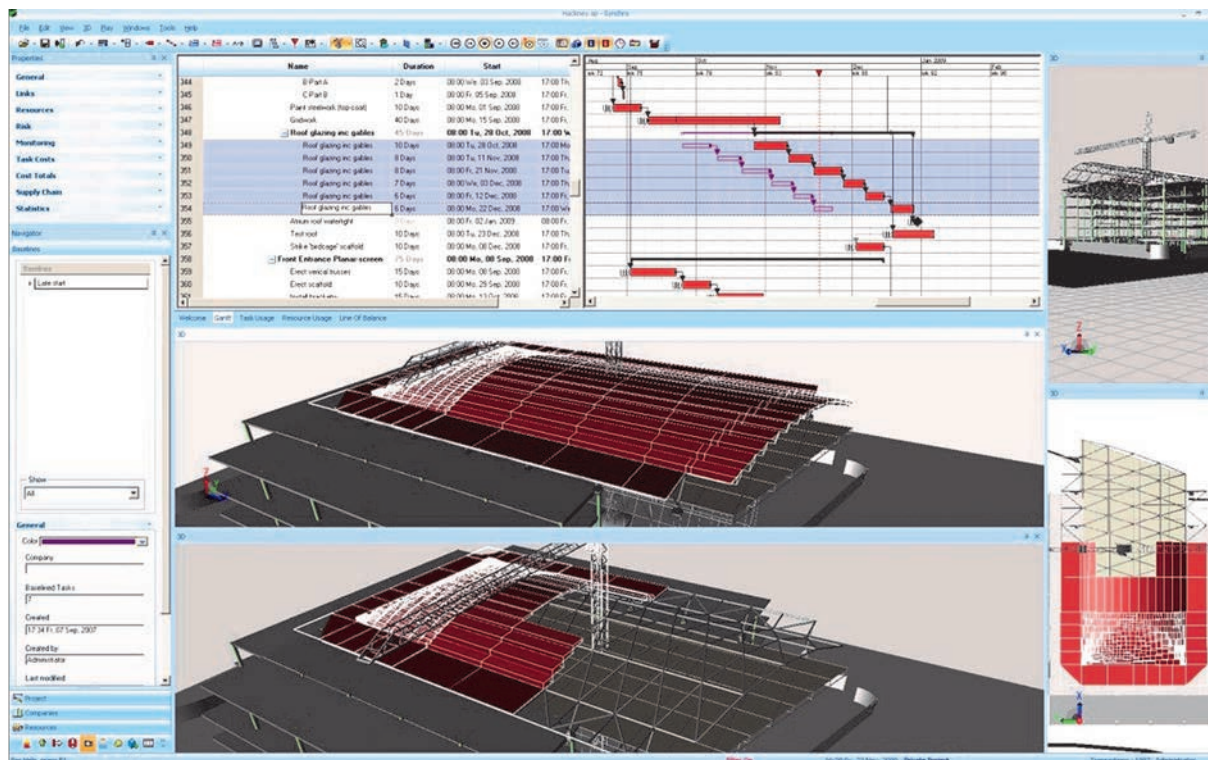


Рис. 1 Экранный снимок системы управления проектами с возможностями 4D анализа, моделирования и планирования

На рис. 1 представлен экранный снимок разрабатываемой системы управления проектами, предусматривающей развитые функциональные возможности для 4D анализа, моделирования и планирования. В верхнем графическом окне экрана проектный план отображается в традиционном представлении дерева композиций работ и диаграммы Ганта. В правом и нижнем окнах этот же план визуализируется в виде соответствующих пространственно-трехмерных сцен проекта. Анимация подготовленного таким образом плана позволяет смоделировать процессы реализации проекта и выявить возможные конфликты, обусловленные пространственно-временными ограничениям и технологическими регламентами. Для полноты визуального анализа предоставляются возможности масштабирования и позиционирования вида сцены в каждом отдельном окне просмотра.

В задачах 4D верификации пространственные ограничения специфицируются аналогично тому, как определяются временные ограничения и отношения предшествования в традиционных постановках календарно-сетевом планирования. В качестве подобных ограничений рассматриваются условия пространственной локализации ресурсов на проектной площадке, их возможные перемещения, технологически допустимое взаимное расположение объектов (например, обязательное расположение одного объекта внутри или снаружи другого), наличие твердых границ, необходимость учета действия гравитации и т.п. Для решения задач верификации применяются известные методы компьютерной графики и вычислительной геометрии, в частности, методы обнаружения коллизий. Перманентное применение методов в каждой точке календарно-сетевого графика приводит к высоким вычислительным затратам, поэтому для локализации потен-

циальных нарушений и минимизации общего числа проверок привлекается конструктивный принцип пространственно-временной латентности.

Не менее содержательной представляется задача 4D планирования проектов с учетом всего набора заданных ограничений. Решение подобной задачи может осуществляться в рамках распространенной концепции обобщенного ресурса. В качестве начального приближения принимается решение, найденное традиционными методами. Затем проводится верификация проектного плана и работам, участвующим в выявленных пространственных конфликтах, последовательно назначаются более поздние сроки исполнения. Процедура верификации и коррекции повторяется до окончательной ревизии календарно-сетового графика, удовлетворяющего всем заданным ограничениям при условии их непротиворечивости. В ряде случаев решение задачи подразумевает комбинаторный поиск плана, который определял бы не только график работ, но и связанные с ним размещение и перемещение ресурсов. В прикладном отношении подобная задача эквивалентна поиску допустимых и оптимальных способов реализации проекта, например, монтажа и обслуживания высокотехнологичного оборудования.

Наконец, конструктивной является задача визуального контроля работ непосредственно на проектной площадке. С этой целью проектный план документируется в виде серий изображений, видеопрезентаций или сценарных моделей в представлении систем компьютерной графики и виртуальной реальности. Затем иллюстрационные материалы, оформленные в виде электронных руководств, сравниваются с реальными изображениями, полученными в результате оперативной видеосъемки проектной площадки, обеспечивая тем самым возможность наглядного контроля и мониторинга работ. Процедура сравнения может осуществляться как непосредственно оператором, так в автоматическом режиме с использованием известных методов распознавания образов и реконструкции пространственно-трехмерных сцен. Отметим, что для автоматического контроля достаточно сопоставлять пары изображений, полученных путем непосредственной съемки проектной площадки и в результате растеризации виртуальной сцены проектного плана. При более доверительном анализе может быть использована и частичная реконструкция геометрических моделей по заданным изображениям проектной площадки.

Решение рассмотренных задач пространственно-временного анализа, моделирования и планирования при реализации актуальных проектов способно, по мнению авторов, обеспечить:

- улучшение координации и взаимодействия сторон в результате выработки единой визуальной картины проекта;
- своевременное выявление и исключение ошибочных ситуаций, обусловленных пространственно-временными и технологическими факторами, уже на этапе планирования;
- эффективный мониторинг и контроль проводимых проектных работ с использованием электронных руководств и детального иллюстрационного материала.

В настоящее время ведутся работы по программной реализации рассмотренных математических методов в составе системы управления проектами. Развитые функциональные возможности для 4D анализа, моделирования и планирования обуславливают внедрение системы в такие индустриальные области, как архитектура и строительство, машиностроение, судостроение, нефтегазовый комплекс, транспорт.

Литература

1. Melin, J.W. & Whiteaker, B. 1981. Fencing a bar chart. *Journal of the Construction Division, Proceedings of the American Society of Civil Engineers* 107(CO3): 497-507.
2. Meredith, J.R. & Mantel, S.J. 1995. *Project management, A managerial approach*, 3d Ed. New York: John Wiley and Sons, Inc.
3. Moder, J.J. & Phillips, C.R. & Davis, E.W. 1983. *Project management with CPM, PERT and precedence diagramming*, 3d Ed. New York: Van Nostrand Reinhold Company.
4. Semenov V.A., Alekseeva E.V., Morozov S.V. *Virtual Construction Using Map-Based Approach*. // *Proceedings of X International Conference on Computing in Civil and Building Engineering*, Weimar, June 02–04, 2004, ISBN 3-86068-213-X, pp. 254-255.
5. V.A. Semenov, S.V. Morozov, H. Jones, A.V. Semenova. *Achieving Schedule Concurrency & Consistency in Major Collaborative Projects*. // *Proceedings of XII International Conference on Computing in Civil and Building Engineering*, 2008, Beijing, China, October 16-18, ISBN 978-7-302-18670-0, pp. 178-179.

Коллективная инженерия на основе визуальных моделей

Семенов В. А., Морозов С. В., Энкович И. В.

Институт системного программирования РАН, e-mail: step@ispras.ru

Обсуждаются проблемы использования визуальных моделей в приложениях коллективной инженерии. Отмечается роль методологий моделирования и связанных с ними стандартов визуальных нотаций. Мотивируется целесообразность привлечения технологий оптимистической репликации и семантической реконсильации для организации эффективной коллективной работы. В качестве приложений рассматриваются задача программной инженерии на основе UML моделей и задача управления проектами с использованием диаграмм Ганта.

В настоящее время средства и системы визуального моделирования получили широкое распространение в разнообразных дисциплинах инженерии. Как правило, они реализуют стандартные визуальные нотации и лежащие в их основе методологии решения прикладных задач. Благодаря этому участники совместной деятельности легко находят между собой общий язык и могут эффективно взаимодействовать в ходе работы над единым проектом. Например, современные системы программной инженерии реализуют актуальную методологию программной разработки MDA на основе универсального языка моделирования UML [1]. Для моделирования промышленной информации находит применение графический язык EXPRESS-G, являющийся одним из элементов распространенной методологии представления и обмена данными STEP [2]. При анализе и реинжиниринге бизнес-процессов традиционно применяется методология и сопутствующая ей визуальная нотация IDEF [3]. В системах календарно-сетевое планирования широко используются методы CPM и PERT, а также тесно связанные с ними визуальные представления планов и расписаний в виде сетевых диаграмм и диаграмм Ганта [4].

Однако попытки коллективной работы с подобными визуальными моделями наталкиваются на препятствия фундаментального характера. Проведение совместных презентационных сессий, исключающих какие-либо конкурентные изменения моделей, возможно при их централизованном хранении под управлением популярных СУБД и применении традиционных пессимистических

транзакций. Однако организация типовых сессий инженерии, предполагающих продолжительную изолированную работу индивидуальных пользователей, крайне проблематична в силу ограниченного мультидоступа к централизованному представлению моделей.

Привлечение технологий оптимистической репликации позволяет отказаться от необходимой централизации управления информационными ресурсами и обеспечить возможность эффективной одновременной работы пользователей со своими версиями визуальных моделей. Ключевой проблемой при этом оказывается согласование дивергентных реплик, которое бы приводило к корректному и содержательному результату, удовлетворяющему всех пользователей. В настоящей работе проблема согласования решается с помощью оригинального метода семантической реконсильации, получившего распространение в ряде прикладных областей. Детальное описание метода и разработанных на его основе приложений можно найти в работах [5, 6].

Метод сочетает в себе следующие принципы и идеи:

- использование формальных моделей прикладных данных для решения задач сопоставления и нечеткого сравнения дивергентных реплик;
- установление отношений зависимости, порядка и композиции между операциями конкурентных транзакций на основе анализа семантических ограничений модели;
- применение методов логического, полисиллогического вывода для выработки перспективных непротиворечивых

(семантически корректных) и полных (обеспечивающих полноту результирующей транзакции) политик реконсиляции;

- комбинирование сценариев автоматической и интерактивной работы с возможностями лексического и семантического согласования.

Рассмотрим применение метода для решения задач коллективной программной инженерии и коллективного планирования.

В последнее время обозначилась тенденция выполнения масштабных программных проектов с использованием CASE средств, ориентированных на повсеместное использование языка моделирования UML и предусматриваемых им кооперативных диаграмм, диаграмм вариантов использования, последовательностей, классов, состояний, компонентов, размещений. Репликация подобных моделей с целью обеспечения индивидуальной работы программистов над единым проектом не влечет каких-либо трудностей. Однако согласование дивергентных версий, полученных в результате автономной индивидуальной работы, оказывается нетривиальной проблемой. Известные средства лексического слияния не могут быть адекватно применены в силу невозможности удовлетворить сложные семантические правила и обеспечить целостность результирующего представления UML модели, формируемой, например, в разметке документов XMI.

В самом деле, метамодель языка UML определяет несколько сотен типов данных и устанавливает для них широкий набор семантических правил, таких как кардинальность объектных популяций, кратность ассоциативных и агрегативных

связей, уникальность имен в области видимости, отсутствие цикличности в отношениях обобщения/специализации типов, упорядоченность элементов последовательностей, согласованность типов, задание обязательных атрибутов. Целесообразным и естественным в данном случае оказывается применение обсуждаемого метода, гарантированно приводящего к семантически корректным и функционально полным результатам и допускающего полноценное использование модельно-ориентированной методологии MDA.

На рис. 1 представлены версии диаграммы состояний, полученные в ходе коллективной работы. Исходная версия диаграммы изображена на рис. 1 (a), а модифицированные версии, полученные в двух ветвях разработки, на рис. 1 (b), (c). Нечеткое сравнение модифицированных версий относительно исходной приводит к реконструкции журналов транзакций в виде последовательностей операций создания, удаления и модификации визуальных элементов диаграммы. Затем на основе формального анализа семантических ограничений метамодели языка UML устанавливаются отношения зависимости, предшествования и композиции между операциями транзакций. Логический анализ отношений приводит к плану реконсиляции и порождаемой им результирующей версии диаграммы, показанной на рис. 1 (d). Данная версия получается в предположении приоритета первой транзакции при разрешении выявленного конфликта одновременной модификации перехода $FlowLine(Task(i=10) \rightarrow State2)$ и $FlowLine(Task(i=10) \rightarrow State3)$.

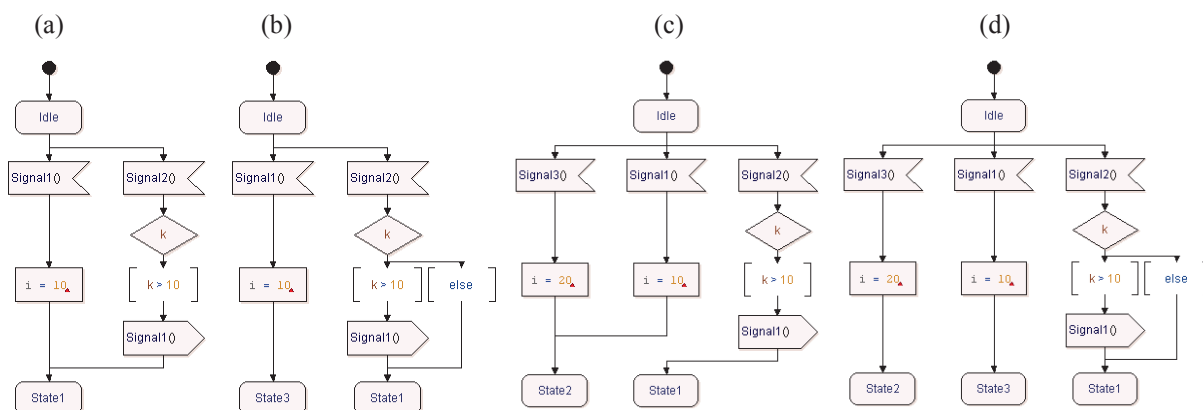


Рис. 1. Дивергентные версии диаграммы состояний UML и результат их семантической реконсиляции.

Обсудим задачу коллективного планирования. Ее постановка предполагает задание проектного

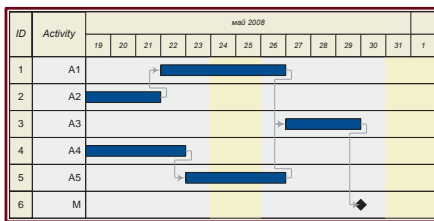
плана в виде декомпозиции работ определенной длительности с заданными отношениями предше-

ствования и временными ограничениями в виде промежуточных вех и сроков реализации проекта. Классический СРМ метод позволяет определить работы, имеющие минимальные временные резервы и влияющие на своевременное выполнение проекта. С этой целью последовательно уточняются допустимые даты начала и завершения отдельных работ, а также устанавливаются для них временные резервы. Анализ существующих модификаций и реализаций СРМ метода показывает, что ни одна из них не может быть применена для решения задачи коллективного планирования.

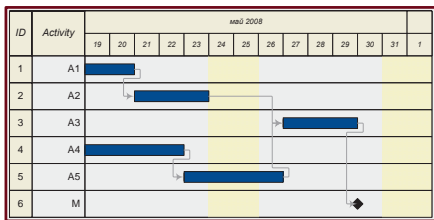
Однако обсуждаемый метод семантической реконсиляции успешно справляется с этой задачей. На рис. 2 изображены реплики календарного плана — базовая версия (а) и модифицированные версии (b) и (c), полученные в ходе параллельной разработки. Непосредственная консолидация конкурентных транзакций приводит к результату, показанному на рис. 2 (d). Данное представление

семантически некорректно, поскольку неизбежно нарушает отношения предшествования между работами $A1$ и $A2$ в силу их цикличности, а также приводит к удлинению критического пути по цепочке работ $A4$, $A5$ и $A3$ и вследствие этого — к нарушению временного ограничения для вехи M .

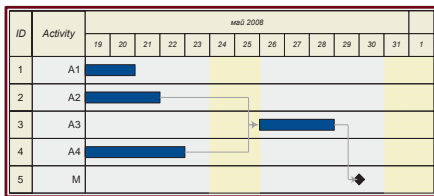
Семантически корректные версии календарного плана, изображенные на рис. 2 (e) и (f), удастся получить с помощью обсуждаемого метода реконсиляции. Конфликты, обнаруженные в результате формального анализа конкурентных транзакций, могут быть разрешены путем альтернативного принятия операций. Разрешив первый конфликт путем принятия изменений, сделанных в первой транзакции, а второй конфликт — принятием изменений из второй транзакции, получим результирующую версию плана на рис. 2 (e). Если изменить приоритеты транзакций при разрешении конфликтов, то сформируется версия плана, представленная на рис. 2 (f).



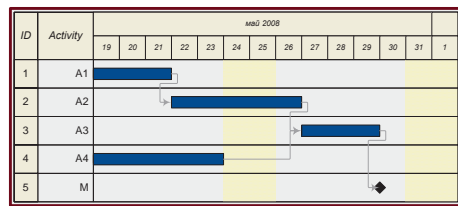
(a)



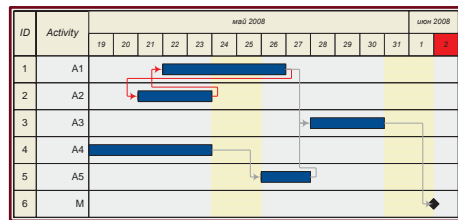
(b)



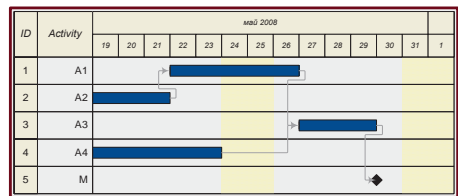
(c)



(d)



(e)



(f)

Рис. 2. Базовая, модифицированные версии календарного плана и полученные результаты их реконсиляции

Таким образом, рассмотренный метод семантической реконсиляции обладает рядом важных достоинств, связанных с гарантированным обеспечением корректности и целостности визуальных моделей, получаемых в результате согласования дивергентных версий, при относительно невысоких вычислительных затратах. Возможности формализованного и содержательного применения к широким классам приложений оптимистической репликации делают его привлекательным, в частности, для реализации перспективных систем коллективной инженерии.

Литература

1. Unified Modeling Language (UML), Version 2.0, <http://www.uml.org/#UML2.0>.
2. ISO 10303-11: 1994, Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual.
3. IDEF Family of Methods: A Structured Approach to Enterprise Modeling and Analysis, <http://www.idef.com>.
4. Moder J. J., Phillips C. R., Davis E. W. Project management with CPM, PERT and precedence diagramming, 3d Ed. New York: Van Nostrand Reinhold Company, 1983.
5. Semenov V.A. Semantics-based Reconciliation of Divergent Replicas in Advanced Concurrent Engineering Environments. // Complex Systems Concurrent Engineering: Collaboration, Technology Innovation and Sustainability, Springer-Verlag, 2007, pp. 557-564.
6. V.A. Semenov, S.V. Morozov, H. Jones, A.V. Semenova. Consistent Reconciliation of Divergent Project Schedules Under Semantic & Functional Constraints. // VIII European Conference on Product and Process Modeling, ECPPM 2008, Sophia-Antipolis, France, September 10-12, 2008, // Product and Process Modeling: eWork and eBusiness in Architecture, Engineering and Construction, Published by Taylor & Francis Group, London, 2008, ed. by P. Morand, R. Scherer, pp. 307-316.

Современный взгляд на структуру систем автоматического построения трехмерных виртуальных моделей по изображениям (обзор)

Юрин Д. В., с.н.с., к.ф.-м.н.

*Институт физико-технической информатики, г. Москва, Протвино,
Московский государственный университет им. М. В. Ломоносова, г. Москва,
тел.: 8 (495) 939-11-29, 8 (926) 175-61-28, e-mail: yurin_d@inbox.ru*

Значительный прогресс в области алгоритмов восстановления 3D по двум и более изображениям [1] обусловлен 1) глубоким развитием методов проективной геометрии применительно к области компьютерного зрения [2] и 2) революционными достижениями в области детектирования характеристических точек на изображениях и сопоставлением их между изображениями [3,4,5], основанными на теории пространства переменных разрешений [6], теориях цветовых [7] и дифференциальных [8,9] инвариантов и созданием дескрипторов таких точек класса SIFT, SURF, GLOH [3,4,5].

Кратко о геометрии зрения. Считается, что камера проективная, т.е. полагая, что начало системы координат совпадает с оптическим центром камеры, координаты точки на изображении (x, y) связаны с трехмерными координатами точки сцены (X, Y, Z) соотношением $x/f = X/Z$, $y/f = Y/Z$, где f – фокусное расстояние, т.е. образом любой прямой линии в трехмерном пространстве будет прямая линия на изображении. Следует иметь в виду, что задача восстановления эпиполярной геометрии (см. далее) весьма чувствительна к радиальной дисторсии (бочке), присутствующей у реальных объективов, поэтому ее коррекция [10,11,12] строго обязательна.

Базовыми понятиями проективной геометрии и трехмерного зрения являются гомография, эпиполярная геометрия, связанная с фундаментальной (или существенной) матрицей (ФМ) и трифокальный тензор. Если две камеры наблюдают один и тот же плоский объект в трехмерном пространстве, то однополные координаты [2,10] $\vec{x}^T = (x, y, 1)$ и $\vec{x}'^T = (x', y', 1)$ соответствующих точек этой плоскости на изображениях связаны соотношением $\vec{x}' = \mathbf{H}\vec{x}$, где \mathbf{H} называется матрицей гомографии и определена с точностью до множителя, т.е. имеет 8 степеней свободы. Ка-

кая бы ни была трехмерная сцена, не деформирующаяся между получением двух изображений (rigid body), образы трехмерных точек на изображениях связаны соотношением $\vec{x}^T \mathbf{F} \vec{x}' = 0$, матрица \mathbf{F} : $rank \mathbf{F} = 2 \Rightarrow \det \mathbf{F} = 0$ называется фундаментальной (ФМ), она тоже определена с точностью до множителя, и, таким образом, имеет 7 степеней свободы. Матрица \mathbf{F} определяется только взаимным расположением камер и их внутренними параметрами и не зависит от трехмерной сцены. Легко видеть, что для каждой точки \vec{x} на одном изображении, это соотношение верно для всех точек \vec{x}' второго изображения, лежащих на прямой линии $\vec{l} = \mathbf{F}^T \vec{x}$, называемой эпиполярной. Такие ограничения называются эпиполярными, ключевым моментом трехмерного зрения является контроль этих ограничений: для недеформируемой сцены, их нарушение является физически невозможным, и алгоритмы не должны рассматривать невозможные ситуации. На каждом изображении эпиполярные линии пересекаются в одной точке, называемой эпиполлюсом. Если на изображениях найдены прямые линии (концевые точки не определены), то для двух изображений не существует подобных геометрических ограничений, но они есть для трех изображений и задаются через трифокальный тензор [2].

С современной точки зрения, восстановление трехмерной сцены должно состоять из следующих шагов.

Детектирование на изображениях характеристических точек. Детектор обязательно должен выполнять три шага:

- детектирование точек на каждом уровне детальности Scale Space, определение масштаба, где особенность проявляется наилучшим образом;
- уточнение масштаба и пространственных координат точки [13], желательна аффинная нормализация окрестности с дополнительным

уточнением координат;

с) построение дескриптора (вектора параметров), который должен быть не хуже, чем у SIFT, SURF, GLOH, Color SIFT [5]. В этом случае наложение ограничений типа малости различий между кадрами, как было в ранних детекторах, не обязательно, зум также может значительно различаться, например в [14] зум различался до 6 раз.

Хорошей идеей является коррекция бочки непосредственно преобразованием координат точек найденных на исходных изображениях, чтобы избежать влияния ресамплинга изображений на точность детектирования точек.

Первичное установление соответствий между точками путем поиска точки, минимально отличающейся по вектору параметров. Следует оставить только надежные соответствия, такие, что расстояние до следующей точки заметно больше найденного (в ~1.5-2 раза). Желательно найти 100-400 соответствий, но не менее 30-50.

Поиск ФМ с помощью алгоритма 11.4 из [2]. Обязательны все три этапа: 7-точечный алгоритм, погруженный в RANSAC, 8-точечный алгоритм по всем инлайерам, нелинейная минимизация. Роль процедуры RANSAC состоит в отсеивании не только ложных соответствий, но и вырожденных конфигураций (эти 7 точек в 3D лежат на поверхности 2-го порядка) или конфигураций, приводящих к низкой точности. Возможные варианты: первоначально искать гомографии Gold Standard алгоритмом 4.3 из [2], это исключает широкий класс вырожденных ситуаций или своевременно о них сигнализирует. Если гомография найдена, ее можно уточнить, применяя ко всей такой области алгоритмы оптического потока [15]. Если найдена только одна плоскость, занимающая значительную часть изображений, вместо 7-точечного алгоритма можно использовать 6-точечный, что благотворно влияет на RANSAC. Если найдено 2 плоскости, то из 2 гомографий можно найти ФМ [16]. Если первичных точечных соответствий было найдено недостаточно, существует алгоритм [17] поиска ФМ по прямым линиям, о поиске и соотношении таких линий (дескрипторы, и их сравнение) см. [18].

Имея эпиполярные ограничения (ФМ), установить соответствия между незадействованными характеристическими точками, сопоставляя по вектору параметров только точки, лежащие в коридоре погрешности [19] вдоль эпиполярной линии. Уточнить ФМ. Применительно к алгоритмам стандартного стерео, фундаментальная матрица позволяет

выполнить ректификацию стереопары [20].

По ФМ можно восстановить 3D координаты точек в единицах базы и ориентацию камер [2,21]. В случае если кадров больше 2, следует путем анализа попарных соответствий найти точки, для которых соответствия удается проследить по большому количеству кадров. Если удалось найти хотя бы 10-20 точек прослеженных по всей последовательности кадров (>4), метод [22] позволяет провести более устойчивое и самосогласованное восстановление координат точек, камер и ориентаций с одновременной оценкой погрешности результата. Если прослежено недостаточно точек или кадров менее 4, то следует воспользоваться тем, что эпиполярные линии на одном кадре, определяемые остальными камерами, в большинстве случаев, будут пересекаться. Это приводит от ограничения «искать вдоль прямой» к ограничению «искать в окрестности пересечения прямых», что позволяет проверить достоверность установленных соответствий и отсеять ошибочные соответствия.

Найденные на шаге 1 характеристические точки, как правило, лежат на локально плоских участках, а не на гранях сцены, поэтому построение сеточной модели по такому облаку 3D точек дает неадекватные результаты. Пользуясь найденными эпиполярными ограничениями, следует добавить виртуальные точки, образованные пересечением линий (edge, см. [1]), найденных на изображениях, с эпиполярными линиями. В качестве дескриптора для таких точек можно использовать подход из [18]. Так как расположение и ориентация камер уже известны, вычисление 3D координат тривиально.

Полученное облако 3D точек следует проанализировать на предмет не выявленных ранее плоскостей, область ими занимаемая может быть расширена путем репроецирования и анализа разностного кадра, а положения плоскостей уточнены [23].

Для неплоских участков трехмерной сцены – построение поверхности путем триангуляции Делоне. Желателен поиск гладких аппроксимаций, хорошей начальной точкой может послужить адаптация подходов [24,25] в применении к фрагментам сцены.

Уточнение границ путем решения задачи сегментации на графах [24]: найден ряд поверхностей, для каждого пикселя каждого кадра найти метку, обозначающую, к какой поверхности его следует отнести, за отнесение соседних точек к различным поверхностям – штраф. При восстановленной сцене и расположении камер, для каждого пикселя одного

кадра выбор поверхности однозначно устанавливает соответствующие ему пиксели остальных кадров. Разность их цвета формирует невязку.

Работа выполнена при поддержке РФФИ, гранты: 08-07-00468-а, 08-07-00469-а, 09-01-92470-МНКС_а, 09-07-92000-ННС_а.

Литература

1. Ссылки на важнейшие работы по 3D и др.: http://imaging.cs.msu.su/~yurin/basic_literature.html.
2. Hartley R., Zisserman A. *Multiple View Geometry in Computer Vision*. – Cambridge University Press, 2004. – 672 p., ISBN: 0521540518.
3. Tuytelaars T., Mikolajczyk K. Local Invariant Feature Detectors: A Survey (http://homes.esat.kuleuven.be/~tuytela/tmp/survey_inv_features.pdf) // *Foundations and Trends® in Computer Graphics and Vision*. – 2008. – V. 3. – No 3, – P. 177 – 280.
4. Mikolajczyk K., Schmid C. A performance evaluation of local descriptors (http://lear.inrialpes.fr/pubs/2005/MS05/mikolajczyk_pami05.pdf) // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2005. – V. 27, – No. 10, – P.1615 – 1630.
5. Burghouts G.J., Geusebroek J.M. Performance evaluation of local colour invariants (<http://www.science.uva.nl/~mark/pub/2009/BurghoutsCVIU09.pdf>) // *Computer Vision and Image Understanding*. – 2009. – V. 113, – P. 48 – 62.
6. Lindeberg T. *Scale-Space Theory in Computer Vision* // Kluwer Academic Publishers, Dordrecht, Netherlands. – 1994. <http://www.nada.kth.se/~tony/earlyvision.html>.
7. Geusebroek J.M., van den Boomgaard R., Smeulders A.W.M., Geerts H. Color invariance (<http://www.science.uva.nl/~mark/pub/2001/GeusebroekPAMI01.pdf>). *IEEE Trans. Pattern Anal. Machine Intell.* – 2001. – V. 23. – No. 12. – P. 1338 – 1350.
8. Schmid C., Mohr R. Local Grayvalue Invariants for Image Retrieval (<http://citeseer.ist.psu.edu/schmid97local.html>) // *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*. – 1997. – V. 19. – No. 5. – P. 530 – 534.
9. Florack L.M.J., ter Haar Romeny B.M., Koenderink J.J., Viergever M.A. Cartesian Differential Invariants in Scale-Space (<http://www.bmi2.bmt.tue.nl/image-analysis/People/LFlorack/Extensions/Flor93c.pdf>) // *Journal of Mathematical Imaging and Vision*. – 1993, – V. 3, – P. 327 – 348.
10. Mohr R., Triggs B. Projective Geometry for Image Analysis // A Tutorial given at ISPRS, Vienna, July 1996 (<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.3924>).
11. Devernay F., Faugeras O. Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured environments // *Machine Vision and Applications*. – 2001. – V. 13. – P. 14 – 24. (<http://devernay.free.fr/publis/distcalib-mva.pdf>).
12. Fitzgibbon A. W. Simultaneous Linear Estimation of Multiple View Geometry and Lens Distortion (<http://www.robots.ox.ac.uk/~vgg/publications/papers/fitzgibbon01b.pdf>) // *Proceedings on CVPR*. – 2001.
13. Brown M., Lowe D. Invariant Features from Interest Point Groups (<http://www.cs.ubc.ca/~lowe/papers/brown02.pdf>) // *In British Machine Vision Conference*. – 2002. – P. 656 – 665. (<http://citeseer.ist.psu.edu/708522.html>).
14. Dufournaud Y., Schmid C., Horaud R. Matching images with different resolutions (<http://citeseer.ist.psu.edu/dufournaud00matching.html>) // *Proceedings on CVPR*, – 2000 (Hilton Head Island, SC, USA). – V. 1, – P. 612 – 618.
15. Zokai S., Wolberg G. Image Registration Using Log-Polar Mappings for Recovery of Large-Scale Similarity and Projective Transformations // *IEEE Transactions on Image Processing* – 2005. – V. 14, – No. 10. (<http://www-cs.engr.cny.cuny.edu/~wolberg/pub/tip05.pdf>).
16. Luong Q.T., Faugeras O.D. Determining the fundamental matrix with planes: instability and new algorithms (<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=CD41FF57989E24F03AEB2D8EF5AAB669?doi=10.1.1.50.8284&rep=rep1&type=pdf>) // *Proceedings on CVPR 1993*. – P.489 – 494.
17. Sagues C., Murillo A.C., Escudero F., Guerrero J.J. From lines to epipoles through planes in two views (<http://robots.unizar.es/data/documentos/06PR-FromLines2Epipoles.pdf>) // *Pattern Recognition* – 2006. – V. 39. – No. 3. – P. 384 – 393.
18. Вологов Д.Б., Юрин Д.В. Предварительное грубое совмещение изображений по найденным на них прямым линиям для построения мозаик, сверхразрешения и восстановления трехмерных сцен (http://imaging.cs.msu.su/pub/2007.JProg.Yurin_Volegov.Draft.ru.pdf) // *Программирование* – 2008, – Т.34, – N.5, – С.47 – 66.

-
19. Csurka G., Zeller C., Zhang Z., Faugeras O.D. Characterizing the Uncertainty of the Fundamental Matrix (<http://citeseer.ist.psu.edu/article/csurka95characterizing.html>) // Computer Vision and Image Understanding. – 1997. – V.68. – No. 1. – P. 18 – 36.
 20. Hartley R.I. Theory and Practice of Projective Rectification (<http://users.rsise.anu.edu.au/~hartley/Papers/joint-epipolar/journal/joint3.pdf>) // International Journal of Computer Vision. – 1999. – V.35. – No. 2. – P. 115 – 127.
 21. Longuet-Higgins H.C. A Computer Algorithm for Reconstructing a Scene From Two Projections // Nature, September 1981. – V. 293, – P. 133 – 135.
 22. Свешникова Н.В., Юрин Д.В. Алгоритмы факторизации: достоверность результата и применение для восстановления эпполярной геометрии (http://www.graphicon.ru/2006/proceedings/papers/fr09_32_Sveshnikova_Yurin.pdf) // Труды конференции Графикон 2006, Россия, Новосибирск, Академгородок, 1 – 5 июля 2006. – С.158 – 165.
 23. Semeikina E., Yurin D. Finding and refinement planes in 3D points cloud obtained under 3D recovery from image set (http://www.graphicon.ru/2008/proceedings/Posters/Poster_15.pdf) // Proceedings of GraphiCon'2008, Moscow, Russia, 2008, P.306.
 24. Birchfield S., Tomasi C. Multiway cut for stereo and motion with slanted surfaces // Proceedings of International Conference on Computer Vision. – 1999, – P. 489 – 495.
 25. Lin M.H., Tomasi C. Surfaces with Occlusions from Layered Stereo. //IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003.

Создание виртуального окружения в тренажерных комплексах

Хураськин И. А., к.т.н., НИИСИ РАН

(Работа выполняется при поддержке РФФИ, грант 09-07-00149-а)

В докладе рассматривается решение задач, связанных с созданием виртуального окружения в тренажерных комплексах. К этим задачам относятся создание адекватного объемного видения виртуального окружения, определение положения человека в пространстве и реализация взаимодействия человека с виртуальными объектами.

При использовании виртуального окружения в имитационно-тренажерных комплексах возникают три задачи:

Создание адекватного объемного видения виртуального окружения.

Определение положения человека в пространстве в реальном режиме времени.

Реализация взаимодействия человека с виртуальными объектами.

1. Технология создания адекватного объемного видения виртуального окружения

Под адекватным объемным видением подразумевается видение 3D сцены в стереорежиме, при котором виртуальные объекты этой сцены, являющиеся моделями реальных объектов, наблюдателю кажутся имеющими те же размеры, что и реальные объекты. Например, пусть некоторый виртуальный объект при стереовизуализации кажется наблюдателю удаленным на расстояние 1 метр. Тогда он должен казаться ему таких же размеров, как и реальный объект, расположенный на таком же расстоянии 1 метр.

Для создания адекватного объемного видения виртуального окружения в тренажерах была разработана специальная технология, которая включает в себя следующие этапы:

Совмещение систем координат реального и виртуального пространства.

Подготовка трехмерной сцены виртуального окружения.

Создание стереопары в соответствии с положением глаз наблюдателя и экрана.

Представление стереоизображения в определенном стереоформате и его вывод.

Первые два этапа технологии выполняются на стадии разработки тренажерной системы. Для совмещения реального наблюдателя и виртуальной сцены необходимо правильно задать системы координат в виртуальном и реальном пространствах. Проще всего считать, что их начала совпадают, оси направлены одинаково и единицы измерения в них также совпадают. То есть мысленно можно совместить оба пространства, если выбрать в них одинаковые единицы измерений.

При подготовке виртуальной сцены в системе трехмерного моделирования (например, 3D MAX) размеры каждого объекта (в единицах системы моделирования) необходимо задавать такими же, как размеры реального объекта (в выбранных единицах измерения реального пространства). Размеры реальных объектов часто можно определить из чертежей, технической документации, фотографий и видеоматериалов. Аналогично, размещать и ориентировать виртуальные объекты следует таким образом, как они должны располагаться в реальном пространстве. Следует также отметить, что при создании 3D сцены важно моделировать свойства поверхности объектов. Шероховатости и неровности объекта, как правило, воспроизводятся с помощью технологий рельефного текстурирования (bump mapping, normal mapping и т.д.). Моделирование поверхности путем только одного наложения текстур (как это зачастую делается при разработке 3D сцен для визуализации в монорежиме) не достаточно, поскольку в стереорежиме такая поверхность будет казаться человеку плоской и нереалистичной.

Остальные этапы технологии выполняются в цикле визуализации виртуального окружения. Третий этап заключается в формировании стереопары с помощью двух виртуальных камер.

Объемное видение оператором виртуального окружения существенно зависит от параметров этих камер. При неверно заданных параметрах человек либо будет видеть искаженное стерео изображение или вообще не ощущать стерео режима. Как известно, параметрами виртуальной камеры являются ее положение, ориентация и пирамида видимости. Для создания адекватного объемного видения виртуального окружения необходимо выполнить следующие требования [1]:

- а) координаты положения виртуальных камер должны быть равны координатам положения соответствующих глаз наблюдателя в реальном пространстве;
- б) каждая камера должна быть направлена перпендикулярно плоскости экрана, на который будет выводиться стереопара;
- в) ребра пирамид видимости камер должны проходить через углы экрана;

Отметим, что эти требования имеют смысл при описанном выше совмещении виртуального и реального пространств.

На четвертом этапе технологии необходимо представить созданную стереопару в определенном стереоформате. Стереоформат – это способ представления изображений стереопары и этот способ тесно связан с устройствами стереовизуализации. Например, формат поочередной смены изображения представляет в четных кадрах визуализации изображение стереопары для левого глаза, а в нечетных – для правого. Для обеспечения попадания в каждый глаз своего изображения используются специальные очки затворного типа или шлемы виртуальной реальности. Формат Side-by-Side подразумевает создание одного общего изображения, в левой половине которого выводится изображение для левого глаза, в правой половине – для правого. Такой формат, как правило, используется поляризационным методом создания стерео с помощью проекторов. Подготовленное изображение выводится видеокарткой, имеющей два видеовыхода, в режиме "горизонтальный спан". Подключенные к двум видеовыходам проекторы визуализируют каждый только свою половину изображения и выводят оба изображения стереопары на один экран, который отражает изображения с сохранением поляризации. Для восприятия стерео используются очки с поляризационными фильтрами.

2. Определение положения человека в пространстве

Для реализации третьего этапа вышеописанной технологии необходима информация о положении глаз человека в пространстве. Получением таких данных занимаются системы трекинга. Для создания виртуального окружения в тренажерных комплексах нами была разработана собственная оптическая система трекинга с использованием светящихся маркеров, которые в некоторой конфигурации жестко крепятся к отслеживаемому объекту (или к голове наблюдателя). Видеокамера, расположенная в определенном месте, производит съемку части пространства, в которой расположен объект. В каждом кадре полученного двумерного изображения специальный программный комплекс выделяет светящиеся маркеры и определяет соответствие между маркерами конфигурации и маркерами на изображении. Далее, используя внутренние параметры видеокамеры и геометрические характеристики конфигураций, комплекс вычисляет координаты маркеров в двумерной системе координат изображения, а затем по ним однозначно определяет их координаты в трехмерном пространстве. Так как конфигурация маркеров жестко связана с объектом, то, тем самым, однозначно определяется его положение и ориентация [2]. Найденное таким образом положение головы наблюдателя однозначно определяет положение его глаз.

Для отслеживания положения головы и кисти руки наблюдателя используются две видеокамеры и шесть разноцветных маркеров. С помощью одной видеокамеры и трех маркеров определяется положение и ориентация головы человека, а с помощью другой видеокамеры и трех маркеров – положение и ориентация кисти руки. Группы (конфигурации) маркеров крепятся к голове и кисти руки человека в определенном порядке, а видеокамеры располагаются таким образом, чтобы наблюдаемые ими маркеры всегда находились в области видимости. Найденные координаты положения и ориентации головы и кисти могут передаваться системе визуализации и системе взаимодействия, которая реализует движение виртуальных объектов при воздействии на них человека. Эта передача осуществляется с частотой не менее 25 раз в секунду, что соответствует реальному режиму времени.



Рис. 1

3. Реализация взаимодействия оператора тренажера с виртуальными объектами

В системах виртуального окружения одной из важнейших задач является обеспечение взаимодействия оператора с виртуальными объектами. Кроме этого, при работе со сложными виртуальными объектами, такими, как пульта управления, оператор должен видеть не только движение виртуальных элементов управления при воздействии на них, но и результат их переключения. Для решения этой задачи нами была разработана технология, основанная на использовании виртуальной кисти. Виртуальная кисть представляется в 3D сцене в виде набора примитивов, приближенно описывающих форму кисти руки человека. Она не отображается в виртуальном пространстве и не имеет визуальных характеристик, но участвует во взаимодействии с другими объектами. Разработанная технология взаимодействия человека с виртуальными объектами включает следующие этапы:

Определение положения кисти руки человека в реальном пространстве.

Определение положения виртуальной кисти в 3D сцене.

Определение момента воздействия кисти на виртуальный объект.

Изменение динамических характеристик виртуального объекта при воздействии на него и реализация логики его работы (создание реакции

на переключение, если этим объектом является элемент управления).

Для определения положения кисти оператора в реальном пространстве используется либо система трекинга, либо компьютерная перчатка. Найденные координаты и ориентация определяют положение и ориентацию виртуальной кисти.

Задача определения момента воздействия состоит в определении пересечения виртуальной кисти с виртуальным объектом. Для быстрого решения этой задачи применяются методы, основанные на использовании ограничивающих объемов в виде прямоугольных параллелепипедов (боксов). Виртуальный объект окружается набором боксов, а решение задачи сводится к определению пересечения боксов в виртуальном пространстве.

Пересечение виртуальной кисти с объектом говорит о том, что между ними произошло столкновение. На четвертом этапе технологии необходимо определить динамику движения виртуального объекта, которую он приобретает в результате этого столкновения. Для обеспечения максимальной реалистичности воздействия на объект результирующее движение вычисляется исходя из законов механики. На движение объекта влияют такие параметры, как тензор его инерции, точка и момент столкновения, скорость и направление движение объекта до столкновения и др.

Что касается виртуальных элементов управления, определение их движения после столкновения с помощью уравнений физики является достаточно сложной и вычислительно-емкой задачей. Поэтому движение виртуальных элементов управления задается более простой моделью, описанной в [3].

При реализации взаимодействия человека с виртуальными элементами управления кроме движения важно обеспечить их функциональность. Каждый элемент имеет свое внутреннее состояние, которое влияет на функционирование управляемого процесса. Состояние элемента управления определяется в основном его положением в пространстве, а переход из одного состояния в другое (переключение), как правило, обуславливается изменением его положения. Поэтому при вычислении положения виртуального элемента проверяется факт его переключения. Если его новое положение соответствует переключению, то выполняются действия, соответствующие этому переключению. Логическое функционирование виртуального элемента управления и условия перехода из одного состояния в другое определяются заранее заданной моделью.

На основе разработанных технологий был построен тренажерный комплекс с использованием виртуального окружения (см. рис. 1).

Литература

1. Хураськин И.А., Михайлюк М.В. Моделирование объемного видения в системах виртуального окружения, Сборник трудов НИИСИ РАН. – 2007. – С. 69-80.
2. Михайлюк М.В., Хураськин И.А. Оптический трекинг с использованием разноцветных маркеров, Сб. научных трудов ИМВС РАН «Высокопроизводительные вычислительные системы и микропроцессоры». – Москва. – 2006. – Вып. 9. – С. 80-90.
3. Хураськин И.А. Взаимодействие с виртуальными пультами управления сложными процессами, // Информационные технологии и вычислительные системы. – 2007 – № 2. – С. 31-41.

Спутниковые сети связи стандарта DVB-RCS с многостанционным доступом MSF-TDMA

Генов А. А., *д.т.н., ФГУП ЦНИИ «Комета»*

Космические аппараты (КА) с приемно-передающей многолучевой антенной (МЛА) и бортовой цифровой платформой (БЦП), реализующей многоканальную конвейерную демодуляцию сигналов земных станций (ЗС), могут стать первым этапом создания КА с мультисервисной бортовой цифровой платформой (МБЦП) стандарта DVB-RCS [4,5] и многостанционным доступом MSF-TDMA. Применение МБЦП в сочетании с МЛА существенно повышает бюджет радиолиний, снижает требования к энергетическим параметрам земных станций (ЗС) и позволяет обеспечивать новые и более качественные услуги связи.

Введение

Практически все действующие в России в настоящее время КА на геостационарной орбите (ГСО) созданы по принципу «прямой ретрансляции» с частным (FDMA) разделением каналов. Несмотря на простоту построения, такие КА имеют ряд существенных недостатков [2,3,6,7]: несанкционированный (пиратский) доступ к ресурсам КА, высокая стоимость центральных ЗС (HUB), невозможность организации каналов прямой связи абонентских ЗС друг с другом, низкую эффективность использования частотно-временного и энергетического ресурса КА в сетях связи стандарта DVB-RCS.

Основными потребителями ресурса таких КА являются, как правило, крупные корпоративные пользователи, имеющие собственные дорогостоящие центральные ЗС. Стандарт DVB-RCS с многостанционным частотно-временным доступом (MF-TDMA) в сетях спутниковой связи с «прямой ретрансляцией» реализуется наземной программно-аппаратной платформой DVB-RCS в составе центральных ЗС (HUB) [6,7].

МБЦП в сочетании с МЛА позволяет реализовать стандарт DVB-RCS с многостанционным пространственно-частотно-временным доступом (MSF-TDMA), разместив программно-аппаратную платформу DVB-RCS на борту КА. Ресурс КА с МБЦП и МЛА становится доступным любому зарегистрированному в сети DVB-RCS пользователю, исключается пиратский доступ к КА, существенно повышается бюджет радиолиний, снижаются требования к энергетическим параметрам

земных станций (ЗС), обеспечивается возможность организации прямой связи абонентских ЗС друг с другом.

1. КА с приемно-передающей МЛА и БЦП, реализующей многоканальную конвейерную демодуляцию сигналов ЗС.

Типовая зона обслуживания действующих в РФ связных КА на ГСО имеет существенно вытянутую с Запада на Восток форму. Ниже, на Рис.1 представлена зона обслуживания КА «Ямал-200» в орбитальной позиции 90E., которая представляет собой вытянутый с Запада на Восток фрагмент глобальной зоны обслуживания КА на ГСО.

Такие зоны обслуживания могут реализовываться достаточно простыми гибридными МЛА с линейным расположением парциальных лучей и шириной луча порядка 5-6 градусов. Причем МЛА по приему и передаче дает более равномерное покрытие зоны обслуживания при существенном выигрыше стволов КА по добротности и ЭИ-ИМ.

С целью снижения затрат на модернизацию действующих связных КА на ГСО предлагается использовать в приемно-передающем тракте для обеспечения покрытия территории РФ гибридные МЛА [1,2] с фиксированной ориентацией в пространстве четырех линейно расположенных парциальных лучей. Ширина парциального луча может быть принята 6-ти градусам.

При линейном расположении парциальных лучей для обеспечения электромагнитной совместимости (ЭМС) между лучами по приему



в каждом створе достаточно выделения двух смежных полос радиочастот $\Delta F1$ и $\Delta F2$. При этом в сети DVB-RCS будет обеспечиваться пространственно (S) частотно (F) временной (T) многостанционный доступ MSF-TDMA.

Каждый парциальный луч гибридной МЛА позволит обеспечить покрытие зоны диаметром

порядка 5000 км и повысить добротность створов КА по приему и ЭИИМ створов КА по передаче на 9 дБ относительно соответствующих параметров глобального луча.

Общая структура приемно-передающего тракта створа КА с БЦП и гибридной МЛА представлена на Рис. 2.

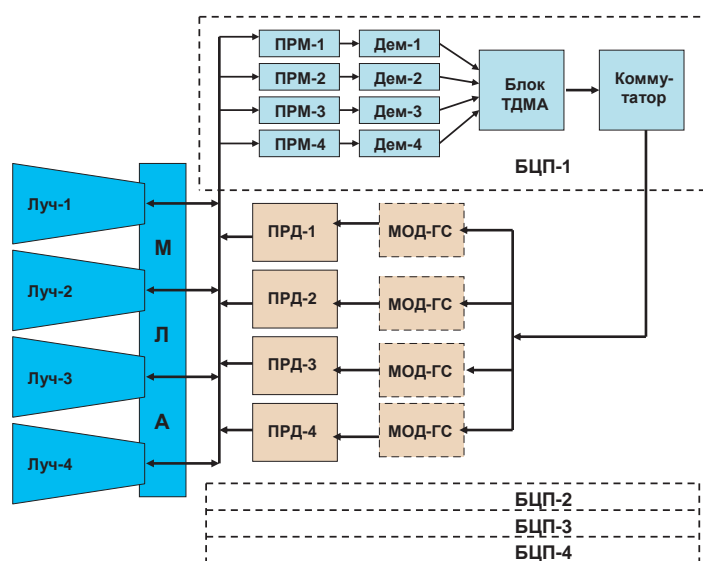


Рис. 2. Общая структура приемно-передающего тракта.

По входу приемной МЛА в каждом из четырех парциальных лучей данного створа реализуется принятый в стандарте DVB-RCS метод многостанционного доступа MF-TDMA. Доступ абонентских ЗС к КА в режиме одночастотной MF-TDMA может обеспечиваться со скоростями от 64 кбит/сек. до 2,048 Мбит/сек. Доступ центральных ЗС к КА в режиме многочастотной MF-TDMA может обеспечиваться со скоростями от 2,048 Мбит/сек. до 28,672 Мбит/сек.

Для сравнительного анализа вариантов реализации стандарта DVB-RCS, в качестве исходных параметров, используются частотно-энергетические параметры стандартного ствола КА «Экспресс-АМ» [8]:

- ЭИИМ ствола в режиме «насыщения» = 50 дБвт;
- добротность ствола $G/T = 3$ дБ⁰/К;
- полоса ствола $\Delta F = 54$ МГц.

В сетях DVB-RCS [6,7], работающих в режиме MF-TDMA без БЦП и МЛА, на линии ЗС – КА – ЦС принимаем:

- кодирование/декодирование по «Витерби», FEC = 3/4;
- кодирование/декодирование Рида Соломона, PC = 47/51;
- метод модуляции/демодуляции - QPSC.

На линии ЦС – КА – ЗС принимаем соответственно:

- кодирование/декодирование по «Витерби», FEC = 7/8;
- кодирование/декодирование Рида Соломона, PC = 47/51;
- метод модуляции/демодуляции - 16PSC.

В сетях DVB-RCS, работающих в режиме MSF-TDMA с МЛА и БЦП, на линии ЗС (ЦС) – КА принимаем:

- кодирование/декодирование по «Витерби», FEC = 3/4;
- кодирование/декодирование Рида Соломона, PC = 47/51;
- метод модуляции/демодуляции - QPSC.
- На линии КА – ЗС (ЦС) принимаем соответственно:
- кодирование/декодирование по «Витерби», FEC = 7/8;
- кодирование/декодирование Рида Соломона, PC = 47/51;
- метод модуляции/демодуляции – 16PSC.

Для варианта с МЛА и БЦП используются параллельно четыре одновременно работающих ствола, БЦП каждого ствола КА имеет 4 приемные линейки с полосой ΔF_1 или ΔF_2 , где $\Delta F_1 = \Delta F_2 = 27$ МГц, а добротность приемных линеек $G/T = (3 + 9) = 12$ дБ⁰/К.

Соответственно каждый блок БЦП включает 4 линейки групповых конвейерных демодуляторов (по одной линейке на каждый парциальный луч МЛА) и блок формирования суммарного группового сигнала TDMA. Дополнительные потери за

счет конвейерной демодуляции определяются числом одновременно демодулируемых сигналов.

С выхода блока формирования группового сигнала TDMA данного ствола групповой сигнал поступает на вход синхронно управляемого цифрового коммутатора и далее после кодирования и модуляции на один из четырех ствольных передатчиков КА «Экспресс-АМ» с полосой $\Delta F = 54$ МГц, постоянно работающих в режиме «насыщения» на один из четырех парциальных лучей МЛА.

В итоге, групповой сигнал с выхода каждого блока БЦП (с периодом порядка 5мсек) с помощью синхронно управляемого цифрового коммутатора последовательно коммутируется на один из 4-х передающих парциальных лучей МЛА. Одновременно с коммутацией группового сигнала меняется частота гетеродина и соответственно несущая частота передатчика, работающего на данный парциальный луч.

2. Сравнительный анализ эффективности использования пространственно-частотно-временного ресурса КА с МЛА и БЦП в сетях связи стандарта DVB-RCS

Для принятых выше исходных данных в сетях DVB-RCS, работающих в режиме MF-TDMA без МЛА и БЦП, в стволе КА с полосой 54 МГц [8], может быть организовано 20 обратных каналов от ЗС со скоростью по 2,048 Мбит/сек и прямой канал от ЦС со скоростью 40,096 Мбит/сек.

Таким образом, пропускная способность стандартной сети DVB-RCS в пересчете на один ствол КА «Экспресс-АМ» в дуплексном режиме составит 40,096 Мбит/сек.

Для сетей DVB-RCS, работающих в режиме MSF-TDMA с МЛА и БЦП, в каждом луче по приему в режиме MF-TDMA в полосе 27 МГц [8] может быть организовано 14 обратных каналов от ЗС и ЦС со скоростью 2,048 Мбит/сек.

Соответственно суммарный групповой сигнал TDMA на выходе БЦП каждого ствола будет иметь информационную скорость 114,688 Мбит/сек. и может быть передан в полосе 54 МГц [8].

В итоге, пропускная способность сети DVB-RCS (в пересчете на один ствол КА «Экспресс-АМ»), работающей в режиме MSF-TDMA с МЛА и БЦП, в дуплексном режиме составит 114,688 Мбит/сек., что в 2,86 раза выше относительно стандартной сети DVB-RCS.

Сравнительная оценка энергетики двух вариантов показывает, что в радиолинии ЗС – КА ре-

жим MSF-TDMA с БЦП и МЛА дает выигрыш ($W_{зс-ка}$) равный:

$$W_{зс-ка} = P_з + P_{мла} + P_{кон} = 7\text{дб} + 9\text{дб} - 8,45\text{дб} = 7,55\text{ дб},$$

где: $P_з$ – стандартный запас в радиолинии ЗС – КА для режима «прямой ретрансляции»; $P_{мла}$ – выигрыш за счет приемной МЛА; $P_{кон}$ – проигрыш за счет конвейерной демодуляции сигналов ЗС и ЦС в БЦП.

Соответственно в радиолинии КА – ЗС режим MSF-TDMA с БЦП и МЛА дает выигрыш ($W_{ка-зс}$) равный:

$$W_{ка-зс} = P_{нас} + P_{мла} + P_{ск} = 3\text{дб} + 9\text{дб} - 4,56\text{дб} = 7,44\text{ дб},$$

где: $P_{нас}$ – стандартный выигрыш за счет перевода передатчика КА из «линейного» режима в режим «насыщения»; $P_{мла}$ – выигрыш за счет «зонального» обслуживания в передающей МЛА; $P_{ск}$ – проигрыш за счет увеличения в 2,86 раза суммарной групповой скорости сигнала TDMA на выходе передатчика КА.

3. Заключение

Приведенные выше результаты показывают, что даже минимальная обработка сигналов ЗС на борту КА (конвейерная демодуляция) в сочетании с введением гибридных приемно-передающих МЛА позволяют существенно повысить эффективность использования частотно-временных и энергетических ресурсов современных КА связи.

К сожалению, пока остаются за кадром вопросы практической реализации полученных результатов для отечественных КА, хотя вопросы создания перспективных КА с МБЦП стандарта DVB-RCS уже не однократно обсуждались [4,5] и нашли практическую реализацию в ряде зарубежных проектов [9,10].

Следует отметить, что для практической реализации указанных выше предложений наибольшую сложность представляет создание быстродействующих цифровых коммутаторов, являющихся ключевым элементом БЦП. Необходимо создание практически абсолютно надежных (САС порядка 15-20 лет) цифровых коммутаторов на скорости передачи информации до 200-300 Мбит/сек.

Дальнейшее развитие БЦП – «Мульти-сервисные бортовые цифровые платформы» стандарта DVB-RCS в сочетании с гибридными приемно-передающими МЛА, которые могут стать в перспективе технологическим ядром большинства современных спутников связи и вещания.

Список литературы

1. А. Генов и др. «Гибридные МЛА», Авторские свидетельства №236135, №236179, Москва, 1986.
2. A. Genov, "The conception of constructing FDMA telecommunication system", "Forum of the IT Proceedings", Spain, 1988.
3. A. Genov, N. Ivanchuk, "The conception of constructing global spread-spectrum CDMA mobile telecommunication "Global-SS" system", "Forum of the ITA Proceedings", Moscow, 1997.
4. А. Генов и др. «Бортовые цифровые платформы – технологический прорыв в повышении эффективности спутников связи и вещания», «Broadcasting», №3, Москва, 2002.
5. А. Генов и др. «Мультисервисные БЦП – технологический прорыв в повышении эффективности ССС», НТК «К 75-летию академика В.А. Мельникова», Москва, 2003.
6. "DVB-RCS - Product Description", EMC TECHNOLOGIES, Канада, 2004.
7. "Sky Edge - Product Description", GILAT, Израиль, 2004.
8. «Регламент», ФГУП «Космическая связь», Москва, 2004.
9. "Euro Sky Way", <http://www.euroskyway.it>, 2003.
10. "Astro Link", <http://www.Astrolink.com>, 2003.

SatExpress – пилотный проект двустороннего спутникового широкополосного доступа

Сургуладзе М. Ш., доктор инфотелекоммуникаций

НИИСИ РАН, sfm12@mail.ru

В середине 1980-х годов, когда Intelsat, а чуть позже Eutelsat стали предлагать услуги VSAT (Very Small Aperture Terminal – спутниковая станция с антенной малого диаметра), связисты быстро сформировали рынок телефонных услуг для труднодоступных регионов. Настоящий бум наступил тогда, когда VSAT-технологии начали применять для оказания современных информационных услуг – передачи данных, специальных корпоративных приложений, доступа в информационные сети и к удаленным информационным массивам. Победное шествие VSAT-технологии сопровождалось не менее победоносным шествием (распространением) Интернет-технологии. Тотальная IP-зация информационных процессов привела к реализации современных инфотелекоммуникационных технологий, в том числе интерактивных спутниковых систем на основе VSAT. Появился симбиоз Интернет и VSAT. Таким образом, VSAT-технологии помогают в преодолении не только проблемы «последней мили», но и «последнего дюйма» в распространении Интернета.

В 90-ые годы 20-го столетия усилиями ученых, конструкторов, инженеров и финансистов, представляющих Европу, Россию, США и Израиль, успешно выполнялся пилотная часть грандиозного проекта под девизом «Always on Broadband Anywhere». Проект предусматривает создание и развитие широкополосных спутниковых сетей, решающих практически все существующие задачи для реализации современных инфотелекоммуникационных систем: организация магистральных каналов, кэш-сервис, односторонний (комбинированный) или двусторонний (интерактивный) доступ в Интернет, «полоса по требованию», конвергенционные компьютерно-телевизионные технологии, мобильное телевидение и многое другое.

Некоторые результаты научно-технических и опытно-конструкторских работ данного проекта стали доступны пользователям в экспериментальном порядке уже с января 2001 года в качестве

инфотелекоммуникационной технологии под названием «satXpress» – high speed asymmetric secure two way IP-services via satellite (ESA-project AO/1-3650 (Multimedia Satellite Services Pilot Operations Contract No ESTEC 14520/00/NL/DS)). Эксперименты, проведенные на низкоскоростных широкополосных широкополосных принципах передачи информации, оказались успешными, в том числе на территории России, в частности сетевых операционных центрах в Главном здании РАН, а также на площадях Математического института РАН и НИИ «Дельта». При этом использовались транспондеры Ku-диапазона геостационарных спутников «Горизонт 11В» (СССР-ПАКА) и «SESAT 36E» (ESA). Были достигнуты скорости до 6Мбит/с в прямом направлении в канале спутник-земля и 128Кбит/с в обратном направлении при обмене информацией в формате MPEG-2 с использованием DVB-стандарта и IP-протокола. Начиная с 22 ноября 2002 года имеется возможность доступа со скоростями до 8Мбит/с в прямом и до 384Кбит/с в обратном направлениях через спутник для массового использования провайдерами.

В указанном пилотном проекте принимали участие научные работники учреждений РАН и НИИ «Дельта» на основании гранта Еврокомиссии по науке и технологиям. Научно-технические результаты по созданию первого поколения высокоскоростных спутниковых терминалов передачи данных были успешно внедрены в Европе и на территории бывшего СССР в Литве, Таджикистане и Грузии в 2002-2005 годах.

Однако научная мысль не останавливается на достигнутом, технический прогресс неумолим и неотвратим. Научно-исследовательские и опытно-конструкторские работы продолжаются. Очередная задача заключается в повышении пропускной способности спутникового магистрального канала до 50Мбит/с в прямом и до 2Мбит/с в обратном направлении. Кроме того, в процессе дальнейших исследований предстоит решать серьезные задачи по повышению степени защищенности

(безопасности) передачи информации путем реализации новейших криптографических алгоритмов и методов шифрования для использования данной инфотелекоммуникационной системы в военных, дипломатических, банковских и других корпоративных целях. Развитие будущих спутниковых инфотелекоммуникационных систем пойдет по пути повышения пропускной способности телекоммуникационной среды (скорости обмена информацией) и безопасности (криптографической защиты информации от несанкционированного доступа) сетей. Итак, беспроводной доступ, скорость и безопасность - 3 кита, чьи плечи выдержат конкуренцию.

Совет директоров всеобщее известной компании «Eutelsat» принял решение о развитии спутниковых сетей с целью построения перспективных инфотелекоммуникационных систем на их базе. Подобные решения появились и в России в планах обновления отечественной спутниковой группировки. Это позволит поддержать развитие услуг спутниковой связи и повысить их качество на уровень требований Интернет-2. Например, ведется разработка, запуск и эксплуатация специальных спутников для поддержания интерактивного широкополосного сервиса с помощью 16-и транспондеров с полосами 36МГц и 4-х транспондеров с полосами 108МГц, предназначенных для обратного канала. На спутниках данной се-

рии предполагается освоить Ка-диапазон. Это – шаг на пути к успешной конкуренции с проводными магистральными средствами связи.

Безусловно, освоение нового Ка-диапазона наталкивается на технические проблемы. Прежде всего, это еще большая, чем в Ku-диапазоне, зависимость надежности связи от погодных условий. Решение проблемы – в наличии достаточного энергетического запаса в сигнале со спутника (что не всегда возможно) на случай неблагоприятных погодных условий и относительно новых для спутниковых сетей методов приема сигнала. По некоторым данным освоение Ku-диапазона показало, что для надежной связи необходимо иметь энергетический запас в 10дБ (вместо 4-6дБ, закладываемых в настоящее время). При прочих равных условиях для Ка-диапазона даже запас в 20дБ не обеспечивает надежную связь. Поэтому при создании сетей этого диапазона используют так называемый разнесенный прием (хорошо известный в системах связи КВ-диапазона). На расстоянии 20-30км располагается второй антенный пост для обеспечения «разнесенного» приема на случай неблагоприятных погодных условий на первом антенном посту. Так что, по-видимому, проблема решаема. Кроме того, важным вопросом массового использования двунаправленных спутниковых систем является обеспечение приемлемых габаритов

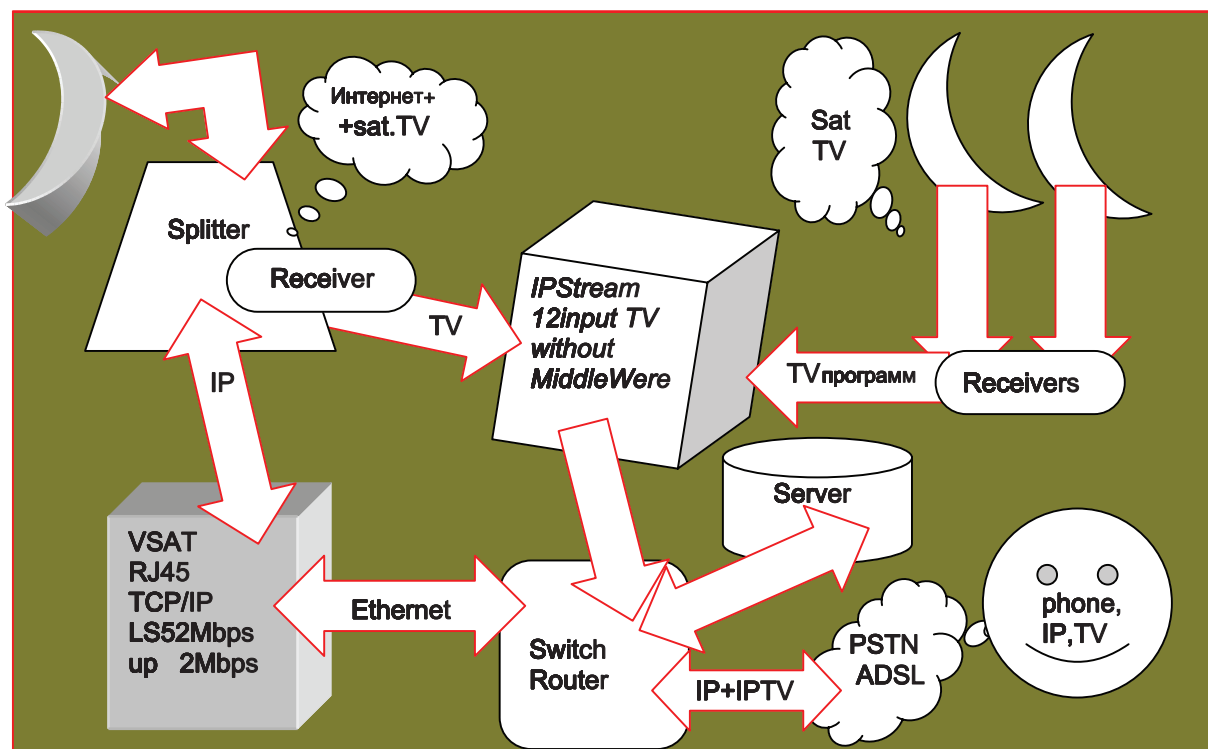


Рис. 1. Инфотелекоммуникационный мультисервисный узел связи для абонентов Triple B1.0-07.

и приемлемой излучаемой мощности передатчика абонентского терминала. Снизить мощность передатчика можно при использовании энергетически эффективных помехоустойчивых кодов. В обратном канале стандарта DVB-RCS предусмотрено использование сравнительно недавно открытых и энергетически эффективных турбо-кодов с возможностью итеративного декодирования. В прямом канале стандарт DVB предусматривает использование сверточных кодов с декодированием по алгоритму Витерби. В результате при антенне с диаметром 0.66м и мощности передатчика 0.5Вт скорость передачи информации в обратном канале может достигать 256Кбит\с.

На рис. 1. в качестве примера автономного узла связи с использованием космических аппаратов приведена схема инфотелекоммуникационного мультисервисного узла связи для абонентов Triple B1.0-07.

Литература

1. Варгаузин В. А., Протопопов Л. Н. «Турбо-коды и итеративное декодирование: принципы, свойства, применение», журнал ТММ №4, 2000г
2. Патент РФ №2122396 «Способ передачи и приема информации и устройство для его осуществления».

**Секция 2. Материалы публикуются отдельной книгой
«Фундаментальные проблемы системной безопасности»**

Секция 3

**Исследование подхода снижения потребляемой
мощности нанотранзисторных цифровых
устройств**

Масальский Н. В., к.ф.-м.н.

Научно-исследовательский институт системных исследований РАН, г. Москва.

Анализируется один из возможных подходов оптимизации потребляемой мощности нанотранзисторных КМОП цифровых устройств, который заключается в отключении функциональных блоков от шин земли и питания в тот период, когда их работа не требуется, что приводит к уменьшению или даже отсутствию токов утечки. На примере 32 разрядного сумматора иллюстрируется эффективность данного подхода.

В настоящее время создание СБИС с пониженным напряжением питания и малой рассеиваемой мощностью является важнейшей задачей для проектирования и изготовления современных микропроцессорных систем, аппаратуры для мобильных, телекоммуникационных и навигационных систем. Поиск путей энергосбережения представляет собой нетривиальную задачу и целиком определяется конкретными требованиями к проектируемому цифровому устройству.

При переходе к нанометровым технологиям в энергопотреблении КМОП СБИС преобладающей становится статическая составляющая мощности. Следовательно, необходимо при выборе подходов к снижению энергопотребления учитывать возрастающий вклад статической мощности. Одним из возможных методов к снижению статической мощности является уменьшение напряжения питания вплоть до отключения функциональных блоков от шин земли и питания в тот период, когда их работа не требуется, что приводит к уменьшению или даже отсутствию токов утечки.

Динамическое управление питанием (ДУП) в функциональных блоках позволяет изменять их производительность в зависимости от требований конкретной задачи, что в конечном итоге существенно снижает уровень потребляемой мощности.

Реализацию ДУП следует рассматривать как две взаимосвязанные задачи:

- контроль питания;
- аппаратная реализация схемы управления питанием.

В рамках первой задачи определяются необходимость и возможность варьирования напряжения питания транзисторов того или иного функционального блока. В рамках второй задачи рассматриваются пути аппаратной реализации схем, непосредственно управляющих питанием блоков, которые были определены в первом пункте.

При аппаратной реализации ДУП питания возникает ряд проблем:

- потери в быстродействии при переходе из одного режима энергопотребления в другой;
- значительные аппаратные затраты на дополнительную логику управления питанием;
- для функциональных блоков, имеющих схемы памяти - затраты на перезагрузку хранящихся данных могут быть больше чем экономия при отключении блоков;
- нахождение оптимального баланса между режимами работы схемы;
- рост динамической мощности из-за добавления блоков управления энергопотреблением;
- поиск оптимального решения между энергопотреблением и ростом площади

кристалла и задержки при наличии схем памяти внутри функционального блока;

- требуется разработка следующих специальных блоков:
- контроллер питания;
- цепи отключения питания;
- регистры данных.

Смена значения напряжения питания занимает определенное время, в течение которого работа блока должна быть приостановлена, а текущие состояния сохранены в регистрах. Следует учитывать тот факт, что изменение напряжения питания в первую очередь влияет на помехоустойчивость схем, поэтому независимо от схемотехники логических цепей регистры должны быть максимально помехозащищенными, то есть статическими.

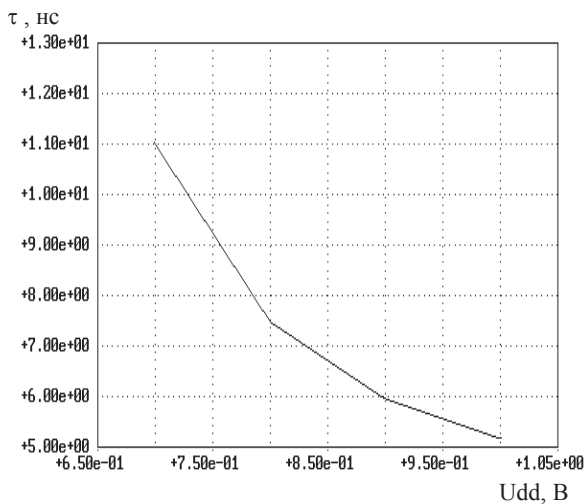


Рис. 1. Зависимость задержки 32-разрядного сумматора от напряжения питания

При модификации схемы сумматора (введения дополнительных регистров на входе и выходе и дополнительного контроллера питания для его отключения) анализировалось влияние оптимизации мощности на площадь кристалла и быстродействие схемы. По результатам моделирования показано, что при использовании предлагаемого подхода при напряжении питания 1 В достигается снижение усредненной мощности сумматора до 35%, при этом динамическая мощность вырастает

В противном случае в процессе перестройки возможна потеря или искажение данных, хранящихся в них. При восстановлении данных необходимо осуществлять контроль правильности их восстановления данных. Эти операции могут значительно увеличить как задержку, так и уровень энергопотребления при переходе из одного режима в другой.

На примере 32-разрядного параллельного сумматора, выполненном на транзисторах по технологии TSMC90 с длиной канала 100 нм численно моделируется данный подход для случая полного отключения устройства от шин питания. Ниже на рисунках приведены результаты моделирования базовых характеристик, проведенные при помощи программы HSPICE.

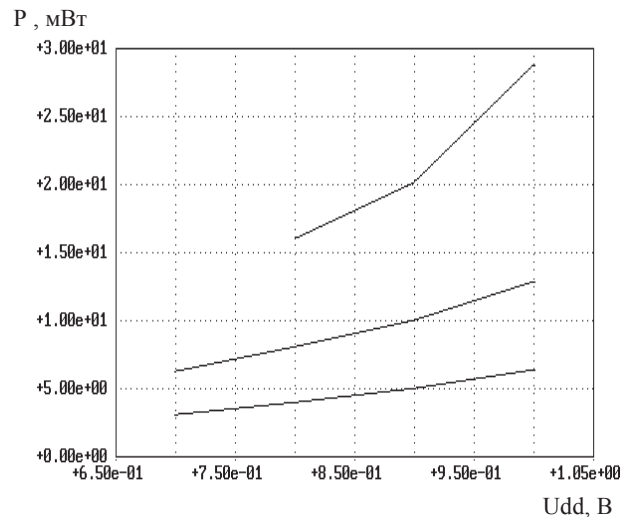


Рис. 2. Зависимость рассеиваемой мощности 32-разрядного сумматора от напряжения питания.

Нижняя кривая – для частоты 0,5 ГГц, средняя – частота 1,0 ГГц, верхняя – частота 2,0 ГГц.

на 12%. Быстродействие схемы в целом уменьшается на 22%. Эффективность данного подхода следует оценивать из соотношения:

$$t_{off} \geq 5(\tau_{in} + \tau_{off})$$

где t_{off} – время, в течение которого сумматор не выполняет операций, τ_{in} , τ_{off} – время необходимое на включение и выключение питания, соответственно.

Статистический метод определения времени начала оптимизаций в динамическом оптимизирующем трансляторе

Гимпельсон В. Д.

ОАО «ИНЭУМ»

Рассматривается задача определения порога начала оптимизаций в динамическом двоичном трансляторе. Предложенная модель использует аппарат математической статистики для вывода соотношений, с помощью которых можно найти статистически оптимальный порог в многоуровневых системах двоичной трансляции.

1. Введение

В последние годы активно ведутся работы по созданию микропроцессоров на базе VLIW архитектуры [2]. Однако при внедрении новых архитектур возникает проблема совместимости со старым x86-совместимым программным обеспечением. Одним из решений этой задачи является динамический комплекс двоичной трансляции [1],[3],[6]. Чтобы такой комплекс был эффективным, он должен включать в себя по крайней мере два уровня: интерпретатор или очень простой компилятор и двоичный оптимизирующий компилятор. Основным требованием для первого уровня является минимально возможное время собственной работы. После того, как количество повторений некоторого участка кода превысит определённую границу, запускается двоичный оптимизирующий компилятор. Он тратит сравнительно много времени на оптимизацию кода и впоследствии всегда используется этот оптимизированный код, скорость работы которого намного выше, чем у неоптимизированного кода. Также возможно, что между этими двумя основными уровнями находятся ещё несколько промежуточных уровней, которые имеют некоторое среднее время преобразования кода исходной платформы в код целевой платформы и в результате вырабатывают некоторый средний по качеству код.

Важной задачей в этом подходе является определение момента, после которого должен начать работать оптимизирующий компилятор, то есть определение границы (порога), по достижении которой код перестаёт интерпретироваться, обра-

батывается оптимизирующим компилятором и после этого работает только оптимизированный целевой код. В данной работе приводится статистический подход к решению этой задачи при заданных начальных условиях (время работы оптимизирующего компилятора, коэффициент улучшения кода и т.д.). Задача сводится к задаче нахождения минимума некоторой функции. Кроме того, найдены необходимые условия достижения минимума.

2. Постановка задачи

Пусть имеется двухуровневый динамический двоично-оптимизирующий комплекс. Первый уровень составляет интерпретатор, а второй – оптимизирующий компилятор. Пусть интерпретатор затрачивает на исполнение одной инструкции исходной платформы в среднем t_n тактов, а оптимизированный код одной команды выполняется в k раз быстрее неоптимизированного (полученного в результате интерпретации), причем на оптимизацию тратится C тактов. Считается заданной также случайная величина ξ , которая определяет количество выполнений произвольной инструкции. То есть, $P(n \leq \xi \leq m)$ – вероятность того, что произвольно взятая инструкция будет выполняться не меньше n раз и не больше m раз. Множеством значений случайной величины ξ является множество натуральных чисел с нулём, однако для простоты дальнейших рассуждений расширим множество значений ξ линейной интерполяцией до множества всех неотрицательных действительных чисел.

Будем считать, что ξ задана своей плотностью $p(x)$. Также положим, что случайная величина ξ имеет конечное математическое ожидание. Пусть код начинает оптимизироваться после того, как некоторая команда исполнилась R раз, при этом зависимость между числом повторений и временем исполнения будет выглядеть следующим образом:

$$\varphi(x) = \begin{cases} t_n x, & x \leq R, \\ t_n R + \frac{t_n}{k} (x - R) + C, & x > R \end{cases}$$

Тогда среднее время исполнения всех повторений произвольной инструкции исходной платформы равняется математическому ожиданию $\varphi(\xi)$. Обозначим эту величину через $A(R) = M(\varphi(\xi))$. Функция $A(R)$ определена при всех R , так как ξ имеет конечное математическое ожидание. Таким образом, задача нахождения статистически оптимального времени начала оптимизаций сводится к нахождению минимума функции $A(R)$.

3. Нахождение статистически оптимального времени начала оптимизаций.

Для нахождения минимума функции $A(R)$, найдём её производную.

$$(1) \quad A'(R) = t_n \left(1 - \frac{1}{k}\right) \cdot \int_R^{\infty} p(x) dx - Cp(R)$$

Приравнявая $A'(R)$ нулю, получаем, что экстремум $A(R)$ достигается в точках удовлетворяющих уравнению:

$$(2) \quad t_n \left(1 - \frac{1}{k}\right) \cdot \int_R^{\infty} p(x) dx - Cp(R) = 0$$

Таким образом, уравнение (2) является необходимым условием статистически оптимального времени начала оптимизаций для двухуровневого двоично-оптимизирующего комплекса.

4. Нахождение статистически оптимального времени начала оптимизаций в многоуровневой системе.

Рассмотрим случай многоуровневой системы. Пусть имеется P уровней оптимизаций в двоично-оптимизирующей системе. Пусть i -тый уровень улучшает код в k_i раз и выполняет оптимизацию одной инструкции за C_i тактов. Пусть также i -тый уровень начинает работать после R_i повторения кода. Положим также $R_0 = 0$ и

$R_{p+1} = \infty$. Тогда время, затраченное на выполнение одной инструкции исходной платформы, равняется:

$$\varphi(x) = \sum_{i=0}^p \frac{t_n}{k_i} X_i(x) + \sum_{i=0}^p C_i Y_i(x), \quad \forall x$$

$$X_i = \begin{cases} 0, & x \leq R_i \\ x - R_i, & R_i < x \leq R_{i+1}, \\ R_{i-1} - R_i, & x > R_{i+1} \end{cases}, \quad Y_i = \begin{cases} 0, & x \leq R_i \\ 1, & x > R_i \end{cases}$$

Посчитаем теперь математическое ожидание времени выполнения одной инструкции (оно определено, так как ξ имеет конечное математическое ожидание):

$$M(\varphi(\xi)) = \sum_{i=0}^p \frac{t_n}{k_i} \left(\int_{R_i}^{R_{i+1}} xp(x) dx + R_{i+1} \int_{R_{i+1}}^{\infty} p(x) dx - R_i \int_{R_i}^{\infty} p(x) dx \right) + \sum_{i=0}^p C_i \left(\int_{R_i}^{\infty} p(x) dx \right) = A(R_1 \dots R_p)$$

Теперь, также как и для двухуровневой схемы, для нахождения статистически оптимальных времён $R_i, i = 1 \dots p$, необходимо найти минимум функции $A(R_1 \dots R_p)$. Необходимым условием минимума функции от многих переменных является равенство нулю всех частных производных. Таким образом необходимое условие минимума имеет следующий вид:

$$(3)$$

$$\frac{t_n}{k_{i-1}} \cdot \int_{R_i}^{\infty} p(x) dx - \frac{t_n}{k_i} \cdot \int_{R_i}^{\infty} p(x) dx - C_i p(R_i) = 0, \quad i = 1 \dots p$$

5. Пример применения полученных результатов.

В этом разделе мы применим полученные формулы для получения статистически оптимального времени начала оптимизаций для четырёхуровневого двоично-оптимизирующего комплекса по формулам, приведённым в разделе 4, на задачах из пакета SPECInt 95, и проанализируем полученные результаты. Положим следующие значения параметров для нашей системы:

$$t_n = 30, k_0 = 1, k_1 = 3, k_2 = 10, k_3 = 50, \\ C_0 = 0, C_1 = 300, C_2 = 5000, C_3 = 100000.$$

В Таблице 1 в первом столбце приведено среднее время исполнения всех повторений одной инструкции при значениях R_1, R_2, R_3 подсчитанных для этой задачи в отдельности, во втором столбце приведено среднее время исполнения

всех повторений одной инструкции при значениях R_1, R_2, R_3 подсчитанное для суммарного профиля всех задач, в третьем отношении этих величин.

Таблица 1. Сравнительная таблица потерь времени для задач из пакета SPECint 95.

	Среднее время исполнения всех повторений одной инструкции	Среднее время исполнения всех повторений одной инструкции при значениях порогов из таблицы 1	Коэффициент потерь времени
099.go_ref	141277	141370	0.9993
124.m88ksim_ref	410222	411994.5	0.9957
126.gcc_ref	3373.79	3915.485	0.8617
129.compress_ref	1167095	1168279	0.9989
130.li_ref	709461	712594	0.9956
132.jpeg_ref	163546	164585	0.9937
134.perl_ref	299592	302195	0.9913
147.vortex_ref	87684.5	87760.4	0.9991
Сред. геометр.			0.9783

Проанализируем полученные результаты. Как видно из Таблицы 1, из общих результатов выделяется задача 126.gcc. Причина состоит в следующем: для всех задач кроме 126.gcc затраты на выполнение лежат в диапазоне от 20 до 65 миллиардов тактов, в то время как 126.gcc исполняется за 500 миллионов тактов, то есть в 50-100 раз быстрее, чем другие задачи. В результате делается предположение, что некоторый код будет ещё долго выполняться, и начинается его оптимизация - в то же время как он исполнение кода длится не очень долго, и в результате время, затраченное на оптимизации высокого уровня, не оправдывается. Однако с другой стороны выполнение этой задачи требует очень мало времени на фоне других задач, и её замедление на 14% компенсируется почти оптимальным (замедления в пределах одного процента) временем исполнения остальных задач.

Список литературы

1. Baraz L. et al, "IA-32 Execution Layer: a Two Phase Dynamic Translator Designed to Support IA-32 Applications on Itanium-based Systems". Proceedings of the 36th International Symposium on Microarchitecture, 2003.
2. Intel. "Intel® Itanium® 2 Processor Reference Manual for Software Development and Optimization". Apr. 2003.
3. Dehnert J.C. et al. "The Transmeta code morphing software: using speculation, recovery and adaptive retranslation to address real-life challenges". CGO, 2003.
4. Muchnick S. S. "Advanced compiler desing and implementation". Morgan Kaufmann Publishers, 1997.
5. Ермолович А.В. Методы повышения производительности двоично-транслирующих систем с аппаратной поддержкой. Диссертация на соискание ученой степени кандидата технических наук, М., ИМВС РАН, 2003.

Реализация программного комплекса, моделирующего вычислительные комплексы с архитектурой SPARC V9

Мешков А. Н.

ОАО «ИНЭУМ»

В докладе приведены принципы построения и основные особенности моделирующего комплекса, используемого при разработке вычислительных средств, базирующихся на микропроцессорах с архитектурой SPARC. Он достаточно полно охватывает всю структуру машины, включая периферийные устройства, и имеет широкие отладочные возможности. Обосновывается выбор объектно-ориентированного подхода к разработке, рассматривается специфика конкретных решений и сферы применения.

1. Введение

Одним из основных методов исследования реализации компьютерной архитектуры является ее моделирование. Большинство средств моделирования можно охарактеризовать как симуляторы пользовательского уровня — они позволяют моделировать поведение пользовательских процессов, эмулируя системные вызовы исполняемые на целевой (моделируемой) системе. В то же время, симуляторы вычислительного комплекса [1,2] практически полностью моделируют целевой компьютер со стадии загрузки, то есть позволяют исполнять системные приложения, такие как программа начальной загрузки и операционная система, как, впрочем, и пользовательские приложения, запущенные после загрузки операционной системы. Поскольку операционная система скрывает большинство подробностей архитектуры компьютерной системы, задача разработки симулятора машины является весьма сложной. Но при этом существенно, что симулятор машины может использоваться для таких целей как анализ поведения задач, основанный на изменении поведения системы (например, сбор статистики по событиям, недоступный на реальной аппаратуре), возможность исполнения программного обеспечения в период, когда аппаратура еще не реализована, разработка и отладка операционных систем и системного программного обеспечения, анализ особенностей различных архитектурных решений и их вклада в изменение производительности системы.

2. Объектно-ориентированный подход к разработке моделирующего комплекса

В реализации программной модели целевой системы был широко использован объектно-ориентированный подход. Несмотря на его достоинства, это решение не является общепринятым при разработке симуляторов [1, 2]. Основной причиной авторы называют проигрыш в скорости моделирования по сравнению с программами, написанными с использованием процедурного программирования. Тем не менее, модульный подход очень удобен при проектировании симуляторов вычислительного комплекса, поскольку отдельные компоненты и устройства системы логично рассматривать как классы.

На рис 1 изображена упрощенная схема симулятора в объектном представлении. Верхним модулем является класс Simulator, отвечающий за разбор ресурсов и предоставление возможности трассировки всем дочерним классам. В зависимости от настроек конфигурации, могут создаваться модели от 1 до 4 микросхем, представляющих собой «системы на кристалле (systems on chip)». Каждая из них содержит 4 процессорных ядра с архитектурой SPARC V9 [3], собственную физическую память (PhysMemory) и может быть связана с южным мостом (South Bridge). Таким образом, в зависимости от конфигурации, возможно моделирование работы вычислительного комплекса с 16 процессорными ядрами. Каждое ядро также имеет блочную архитектуру, в качест-

ве составляющих которой выступают основные устройства процессора — регистровый файл (integer regfile), устройство управления памятью (MMU), устройство выполнения операций с плавающей запятой (FPU), декодер и так далее. Устройства процессора также могут содержать собственные блоки там, где это необходимо.

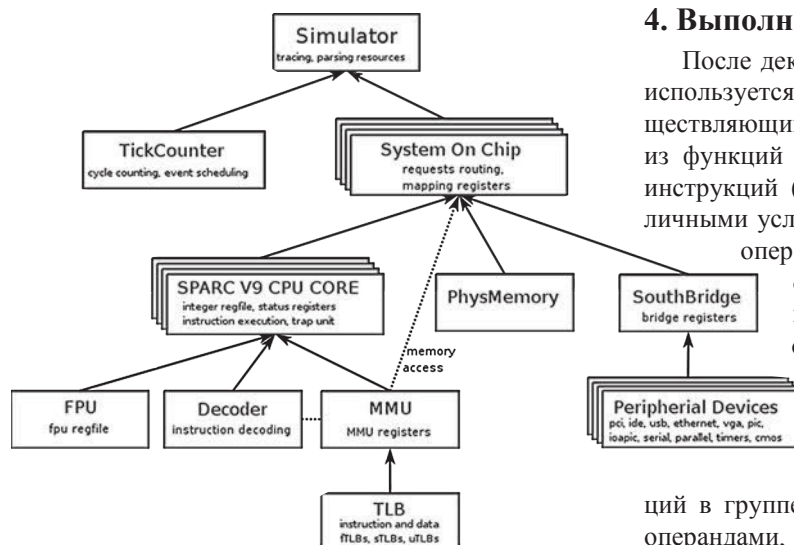


Рис 1. Упрощенная схема моделирующего комплекса архитектуры SPARC V9

С функциональной точки зрения процессорное ядро производит основной цикл выполнения, включающий такие действия как загрузка инструкции, декодирование, чтение операндов, исполнения и так далее. Ниже рассмотрены некоторые особенности их реализации в модели, в том числе те, которые позволили обеспечить приемлемую производительность программного комплекса.

Декодирование инструкций в системе команд SPARC V9 — это преобразование 32-битной инструкции в тип операции, непосредственные операнды или номера регистров, содержащих операнды, и номер регистра, содержащий результат. Декодирование является достаточно трудоемкой задачей, вместе с тем оно должно быть быстрым и точным. Традиционно декодеры производят разбор вплоть до кода операции (в данной модели более 300) [2, 4], попутно заполняя структуры, хранящие значения операндов. Это приводит к появлению большого количества таблиц переходов и, как следствие, чрезмерному разрастанию декодера, снижению читабельности кода и понижению скорости. Вместо этого, в разработанном симуляторе реализован упрощенный предекодер. Команда разбирается только до определения

группы операций — совокупности схожих по исполнению инструкций. Такое разбиение определено на уровне системы команд, в описании инструкций они уже разделены на группы. Это позволяет существенно упростить декодер (групп операций насчитывается около 60) и отложить дальнейший разбор до момента исполнения.

4. Выполнение инструкций

После декодирования номер группы операции используется как индекс в таблице функций, осуществляющих выполнение инструкций. Каждая из функций соответствует определенной группе инструкций (например команды перехода с различными условиями или всевозможные варианты операций сдвига), и в ней происходит окончательное декодирование и исполнение операции. Такой способ организации позволяет заменить несколько сотен функций, многие из которых практически идентичны, несколькими десятками. При этом, в силу однородности операций в группе, различаются только действия над операндами, в то время как сами операнды декодируются едино для всей группы, то есть значительно упрощается и структурируется исходный код.

5. Обработка прерываний

Подсистема прерываний является одной из важнейших в процессоре. Система команд SPARC V9 имеет сравнительно сложную архитектуру, поддерживающую 5 уровней прерываний. В то же время, поведение системы несильно отличается от уровня к уровню (исключением является лишь последний отладочный уровень) [3]. Это, а также необходимость реализации только двух типов прерываний (точных и асинхронных, ибо отложенные прерывания в реализации процессора не предусмотрены), позволяет создать достаточно компактный модуль обработки прерываний. Точные прерывания (исключения) обнаруживаются в момент выполнения инструкции и срабатывают в порядке приоритета. Обработка точных прерываний реализована с помощью класса exception стандартной библиотеки C++. Такой подход позволяет значительно упростить читабельность кода по сравнению с традиционным методом реализации через системные вызовы setjmp/longjmp [2,4] и избавиться от возможной потери производительности, связанной с использованием системного вызова longjmp. Единственное требование —

проверка точных прерываний должна производиться в порядке их приоритетов. Однако, оно не является трудновыполнимым, так как приоритеты соответствуют степени обработки инструкции, то есть с продвижением инструкции по конвейеру они понижаются.

6. Отладочные возможности

К основным характеристикам разработанного моделирующего комплекса, в первую очередь, относится возможность максимально подробной трассировки. Это трассировка состояния симулятора, отображение декодированного кода и выполняемых в соответствии с ним действий, трассировка прерываний и исключений, трассировка изменения состояний регистров и памяти по заданному физическому адресу, печать общей информации о выполнении при выходе из программы. Допускается и выборочная трассировка в заданном диапазоне адресов исполняемого кода или тактов.

Любой из узлов симулятора может быть отлажен путем трассировки каждого из устройств в отдельности. При этом возможна трассировка как отдельных узлов микропроцессора, таких как: устройство управления памятью (MMU), устройство трансляции адресов (TLB), кэши данных и команд 1-го уровня, кэш 2-го уровня и т.д., так и периферийных устройств.

Еще одним полезным инструментом отладки симулятора и приложений является поддержка механизма работы с контрольными точками, содержащими точный слепок системы на момент съема данных о ее состоянии: всех регистров процессора и периферийных устройств, содержимого всех занятых областей памяти и значения любых переменных, оказывающих влияние на поведение системы.

7. Применение

Разработанный симулятор был задействован в ряде задач. В первую очередь, надо отметить формирование направленных архитектурных тестов, проверяющих соответствие реализации процессора специфицированной для него системе команд. При этом тесты и симулятор разрабатывались независимо, что позволило провести начальную отладку симулятора на тестовом пакете. К моменту написания публикации тестовый пакет насчитывал более 1200 направленных тестов, успешно выполняющихся на симуляторе. Вторым принципиально важным применением является разработка генератора случайных тестов, предназначенно-

го для повышения полноты тестирования и покрытия. В третьих, следует отметить создание программы начальной загрузки (boot), выполняющейся с момента старта вычислительного комплекса до передачи управления операционной системе. Одной из задач загрузчика является начальное тестирование и настройка аппаратуры, соответственно, для корректной работы загрузчика на симуляторе потребовалось очень точное соответствие моделируемого железа реальному. И наконец, моделирующий комплекс был в полной мере использован при портировании операционной системы Linux для работы на данной архитектуре. Загрузка операционной системы, как ничто другое, требует особой точности и корректности моделирования не только всех архитектурных особенностей, но и поддержки множества периферийных устройств.

Список литературы

1. Magnusson, P. S., Dahlgren, F., Grahn, H., Karlsson, M., Larsson, F., Lundholm, F., Moestedt, A., Nilsson, J., Stenström, P., Werner, B. (1998), SimICS/sun4m: A Virtual Workstation, in 'Usenix Annual Technical Conference'.
2. Rosenblum, M., Herrod, S. A., Witchel, E., Gupta, A. (1995), Complete Computer System Simulation: The SimOS Approach, IEEE Parallel and Distributed Technology .
3. Weaver, D., Germond, T., (1994), The SPARC Architecture Manual, Version 9, SPARC International, Inc.
4. Zadarnowski, P. (2000), The design and implementation of an extendible instruction-set simulator, BE thesis, School of Computer Science and Engineering, University of New South Wales, Australia.

Разработка планировщика задач реального масштаба времени для ОС Unix

Федотов Д. А., аспирант

ОАО «ИНЭУМ»

Одним из важных аспектов функционирования операционных систем в реальном масштабе времени (РМВ) является планирование процессов, при котором существенно минимизируется время постановки активируемого процесса на процессор. Этот фактор имеет важное значение при доработке Unix-подобных операционных систем для использования в режиме жесткого РМВ [1]. Доработка была проведена в рамках создания программного обеспечения многопроцессорных вычислительных комплексов ВК «Эльбрус-90микро» на базе операционной системы Solaris 2.5.1.

Распределение задач между процессорами

Анализ применений, в интересах которых проводилась доработка, привел к необходимости изолировать процессоры комплекса, на которых исполняются задачи РМВ, от исполнения менее приоритетных задач. Это обеспечивается привязкой РМВ задач (и только их) к выделенным процессорам. Для каждого из них операционная система комплекса создает локальную очередь РМВ процессов. В целях более равномерной загрузки в системе также определена одна глобальная очередь для процессов РМВ, из которой выделенные процессоры могут получить работу в случае ее отсутствия в локальных очередях. При назначении работы определенному процессору диспетчер сначала определяет ее наличие в локальной очереди. Если очередь пуста, то он проверяет глобальную очередь РМВ процессов и, если она не пуста, активирует на данном процессоре ее головной процесс.

Схема детерминированного распределения процессов требует отключить их динамическую миграцию между процессорами комплекса, связанными с общей памятью. Естественно, что при этом повышается вероятность нахождения данных, используемых задачами, в кэш-памяти процессора и устраняется необходимость обеспечения когерентной работы двух процессоров при миграции, которая неизбежно вносит дополнительную задержку в их работу.

Активация процессов

В общем случае переключение процессора от исполнения одной задачи на активацию другой происходит, если исполняемая задача исчерпала выделенный ей квант времени, ожидает ресурс или требует активации более приоритетный процесс. С точки зрения режима РМВ третья ситуация является наиболее критичной, так как согласно штатным механизмам ОС Unix необходимое переключение реально произойдет только после отработки очередного прерывания или системного вызова на данном процессоре, а также - при приходе прерывания системного таймера на любой процессор [2]. Поэтому в данной реализации в планировщик процессов введен механизм, вырабатывающий одновременно с возникновением запроса на активацию более приоритетного процесса прерывание высокого уровня, единственной задачей которого является переключение процессора от текущей задачи.

Приостановка запусков планировщика системным таймером

В режиме РМВ работа процессоров должна быть строго подчинена приоритетному планированию - процесс с меньшим приоритетом не будет активирован, пока более приоритетный процесс не завершится или не заснет в ожидании внешнего события или ресурса, в то же время идущий процесс может исполняться процессором неограниченно долго, пока не появится процесс с большим приоритетом. Этот принцип устраняет необ-

ходимость запусков планировщика системным таймером, которые следует приостановить.

Обслуживание прерываний

Запуску или исполнению РМВ процесса может помешать внешнее прерывание. Если в целевой системе контроллер прерываний позволяет назначать определенные типы прерываний отдельным процессорам, то необходимо обеспечить изоляцию процессоров, на которых исполняются задачи РМВ, от обработки прерываний, не используемых процессами реального масштаба времени. Иначе это выполняется путем программной эмуляции в операционной системе.

При невозможности изоляции процессора от обработки прерываний необходимо минимизировать время, в течение которого процесс РМВ может быть приостановлен для обработки прерывания. С этой целью вводится двухфазная схема обработки прерываний, аналогично используемой в Linux. В первой фазе при задержанном РМВ процессе выполняются ограниченные (кратковременные) действия, после чего вызывается вторая фаза обработки. Во вторую фазу выносятся рутинные процедуры, которые могут быть откачаны вновь активированным РМВ процессом, находящимся выше по шкале приоритетов.

Таким образом, с точки зрения операционной системы различные классы прерываний характеризуются следующей иерархией приоритетов.

Приоритеты	Классы процессов
190-175	Процессы первой фазы прерываний
174-116	Процессы реального масштаба времени
115-100	Процессы второй фазы прерываний
99-60	Системные процессы
59-0	Процессы с разделением времени

Требование работы в жестком режиме РМВ привело к необходимости разделить обслуживание прерывания системного таймера, занимающего самое длительное время, на две части. Процессы первой части, связанные с настройкой систем-

ного времени, запускаются на любом доступном для обработки прерываний процессоре. Вторая часть, ответственная за запуск планировщика и обработку отложенных записей, обязательно исполняется, если существуют готовые для активации отложенные задачи реального времени [3]. Если же таких задач на данный момент нет, то вторая часть будет исполняться только в случаях, если процесс системного таймера не прервал другое прерывание или задачу реального времени.

Однопроцессорные комплексы

В случае однопроцессорного комплекса нельзя отключать процессор от прерываний, но всю остальную схему можно использовать, при этом время установки процесса реального масштаба времени на исполнение будет, в худшем случае, равно времени обработки самого долгого прерывания.

Литература

1. D. Francis, "A Scalable Data Acquisition System Using the Chorus Operating System," Open Bus Systems '05, Zurich, Switzerland, 11-13 October 2005.
2. Вахалия Ю. Unix изнутри // издательский дом Питер 2003. «Классика computer sciences».
3. Федотов Д.А. Оптимизация ОС Solaris для выполнения задач реального времени. // Труды 50-й научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук». 2007. Т. 1. С. 69-71.

Построение многопроцессорного NUMA-сервера на базе процессорного модуля «Эльбрус-3S»

Зайцев А. И., Шерстнёв А. Е.

ОАО «ИНЭУМ»

Рассматривается разработка процессорного модуля «Эльбрус-3S» для построения NUMA-серверов среднего уровня с аппаратной поддержкой когерентности. 4-процессорная система может быть организована без использования дополнительных компонентов путём соединения процессоров. Увеличение числа процессоров требует использования специального коммутающего чипа. В работе приводятся решения основных проблем, связанных с нехваткой пропускной способности межпроцессорных интерфейсов, в 16-процессорных системах.

Можно выделить два основных принципа построения многопроцессорных систем. Первым из них является симметричная многопроцессорная система (SMP), представляющая собой иерархическое дерево с общим коммутающим модулем, в качестве листьев которого выступают отдельные процессоры. Выход на общую память имеется только в коммутающем модуле. В большинстве случаев такие системы не расширяемы по максимальному объёму оперативной памяти, пропускной способности канала памяти и числу процессоров – все основные параметры определяются на стадии создания системы.

Вторым типом является система с неоднородным доступом в память (NUMA). NUMA-система представляет собой сеть определённой топологии из однотипных модулей (процессоров). Каждый процессор имеет встроенный контроллер памяти, обслуживающий часть общей памяти системы. Основным достоинством NUMA-систем является хорошая масштабируемость, а недостатком – неоднородное время доступа в память, а также – увеличение времени доступа по мере увеличения числа процессоров.

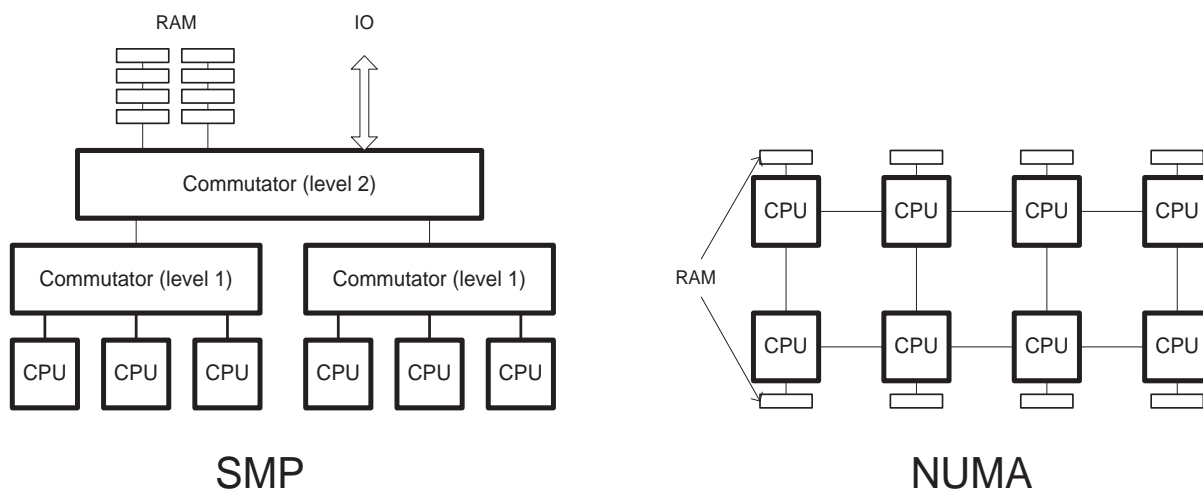


Рис.1. Примеры основных типов организации многопроцессорных систем

Процессорный модуль «Эльбрус-3S» представляет собой систему на кристалле (SOC – System-on-Chip), имеющую в своём составе процессорное ядро «Эльбрус», двухканальный оптимизирующий контроллер памяти и оборудование интерфейсной логики - «северного моста», в дальнейшем обозначаемого как системный коммутатор. Модули «Эльбрус-3S» предназначены для построения много-процессорных серверов типа ссNUMA¹. Процессоры объединяются в когерентную систему через высокоскоростные последовательные каналы межпроцессорного обмена, свя-

зывающие каждый процессор со всеми остальными (Рис. 2).

Приоритетной задачей данной разработки было обеспечение эффективного взаимодействия процессора с оперативной памятью, как по пропускной способности, так и по времени доступа. Это потребовало применения ряда оптимизационных решений, соответственно, в интерфейсной логике и устройствах обработки запросов.

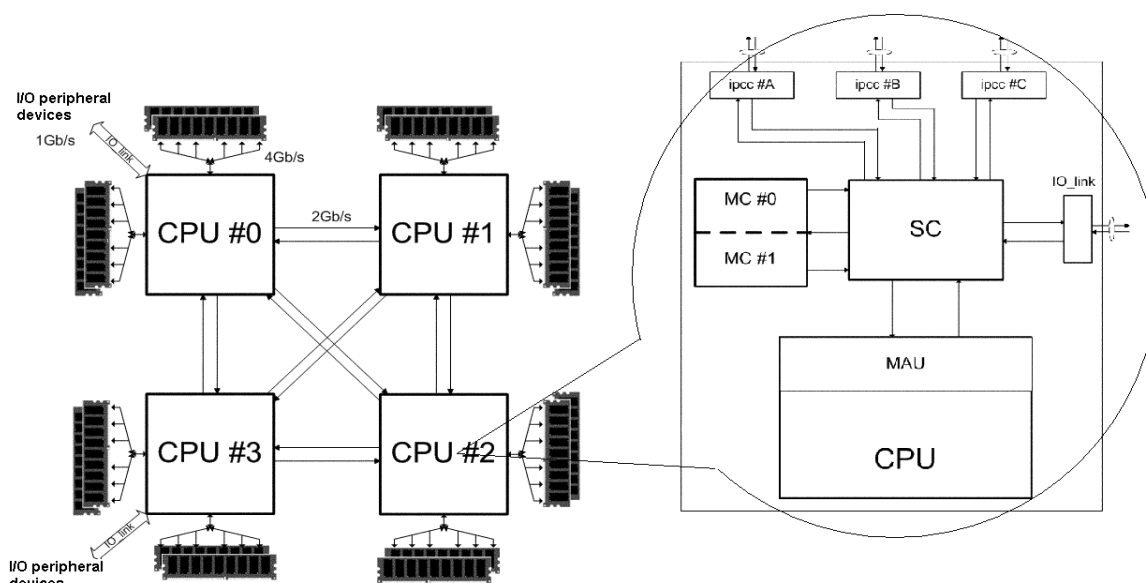


Рис.2. 1-кластерный ВК "Эльбрус-3S": ipcc – контроллер межпроцессорного обмена, MC – контроллер памяти, IO_link – контроллер канала ввода/вывода, SC – системный коммутатор (СК), MAU – устройство работы с памятью, CPU – процессор.

Одним из компонентов, потребовавшим новой разработки, является системный коммутатор (СК). На этот модуль ложатся функции «северного моста», то есть обеспечение обмена между процессорным ядром, локальной оперативной памятью и внешними по отношению к данному процессорному модулю абонентами (другими процессорами, «южным мостом»). СК является ключевым элементом при построении 4-процессорной системы, именуемой в дальнейшем «кластер».

В разработанном системном коммутаторе используются оптимизации, преследующие три основные цели:

- Минимизация времени доступа в память (для операций чтения),
- Повышение производительности конвейера обработки запросов

- Эффективная работа с большим числом абонентов (5).

Когерентность в NUMA-системе «Эльбрус-3S» поддерживается на аппаратном уровне при помощи техники «снупирования». Это означает, что при каждом обращении процессора в память формируются запросы проверки когерентности, направляемые в контроллер кэша каждого процессора. Снупируемый процессор высылает в процессор, пославший запрос, ответ, означающий, что в кэше данного процессора нет модифицированных данных, или содержащий модифицированные данные.

Для исключения чрезмерного усложнения логики СК и избыточности протоколов межпроцессорных каналов, с помощью СК возможно объединение не более четырёх процессоров, то-есть,

¹ Cache coherent Non-uniform Memory Access – когерентная система с неоднородным доступом в память.

создание только одного кластера (см. рис. 2). Для дальнейшего увеличения числа процессоров требуется дополнительный чип – контроллер межкластерного обмена (КМО). Его основным назначением является объединение нескольких кластеров (до 4-х) для создания серверов среднего уровня, то есть когерентных систем с общей памятью с числом процессоров не более 16 (рис. 3). В функции КМО входит трансляция обращений в память удалённых кластеров, а также формирование запросов проверки когерентности для процессоров данного кластера. При взаимодействии двух КМО используется расширенный протокол межпроцессорного обмена.

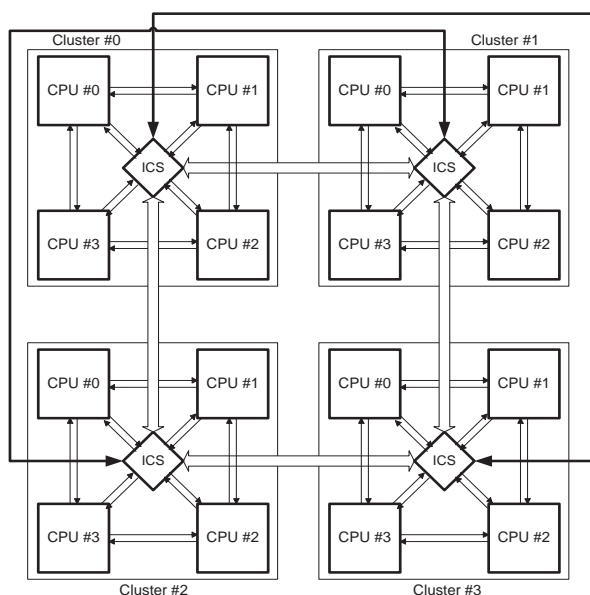


Рис. 3. 4 кластера, объединённые посредством КМО (ICS – Inter Cluster Switch)

Основной проблемой при увеличении числа процессоров в NUMA-системах является квадратичное² по отношению к количеству процессоров увеличение вырабатываемых запросов проверки когерентности и, как следствие, рост величины задержки доступа к данным и падение эффективности пропускной способности каналов межпроцессорного обмена. В качестве решения данной проблемы без усложнения логики процессорного модуля, в частности системного коммутатора, предлагается сократить число запросов проверки когерентности, путём оптимизации работы КМО. Для этого в КМО помимо коммутирующей логики вводятся два устройства: фильтр и справочник. Фильтр оптимизирует доступ к удалённой памяти

то-есть, к памяти, обслуживаемой в других кластерах., справочник – к локальной. Фильтр отслеживает состояние кэш-строк удалённых процессоров, а также содержит идентификатор процессора-владельца, если данные кэшированы. Ниже приводится несколько вариантов выполнения считывания удалённых данных с участием фильтра. Варианты расположены в порядке увеличения времени доступа и загрузки межпроцессорных и межкластерных каналов. При обращении к удалённой памяти КМО определяет при помощи фильтра, что:

Запрашиваемые данные отсутствуют в кэшах процессоров других кластеров, но используются (строка в кэше в состоянии “Modified”) одним из процессоров локального кластера. КМО не формирует обращений в удалённые кластеры, а всем процессорам локального кластера отсылаются запросы проверки когерентности, тип которых соответствует типу исходного запроса. Модифицированные данные из удалённого кластера-владельца данных есть в кэше процессора. КМО направляет в соответствующий кластер запрос проверки когерентности, возвращающий в качестве ответа данные из кэша процессора-владельца данных

Данных нет ни у одного процессора. КМО в этом случае транслирует исходный запрос в кластер, обслуживающий данную ячейку памяти (кластер-владелец), где запрос передаётся в процессорный модуль и, далее, в контроллер памяти.

Модифицированные данные есть в кэше процессора из удалённого кластера. КМО передаёт исходный запрос в кластер-владельца, откуда по справочнику запрос транслируется в кластер, содержащий процессор с модифицированными данными.

Справочник служит для слежения за локальной памятью. В справочнике содержится информация о том, есть ли данные из памяти, принадлежащей кластеру, в кэше процессоров, входящих в удалённые кластеры. Если после проверки оказывается, что данные не кэшированы в удалённых процессорах, это позволяет не выдавать запросы проверки когерентности за пределы кластера. Если же данные есть в кэшах одного или нескольких процессоров из других кластеров, а локальный процессор, выдавший запрос, собирается произвести модификацию данных по этому адресу, то во все удалённые кластеры, содержащие эти данные,

² При наличии в системе n процессоров в каждом такте может быть сформировано n обращений в память, каждое из которых вызовет генерацию n запросов проверки когерентности.

направляются сообщения о фиксации инвалидности рассматриваемого адреса.

Таким образом, в случае обработки локальных данных, на что и ориентирована архитектура NUMA, применение фильтра и справочника в КМО позволяет полностью исключить пересылку между кластерами ненужных запросов (как первичных запросов, так и запросов проверки когерентности), что значительно повышает эффективность многокластерных серверов. При этом со стороны пользователя и программиста сохраняется возможность работы с полным объемом оперативной памяти без каких-либо модификаций существующего программного обеспечения и стиля программирования³.

Литература

1. <http://www.mcst.ru>
2. A. Ahmed et al., AMD Opteron Shared-Memory MP Systems (http://www.hotchips.org/archive/hc14/program/28_AMD_Hammer_MP_HC_v8.pdf).
3. R. Oehler and R. Kota, Horus: Large-Scale SMP for Opteron
4. D.Abts, A.Batanieh, S.Scott, G.Faanes, J.Schwarzmeier, E.Lundberg, T.Jonson, M.Bye, G.Schwoerer, The Cray BlackWidow: A Highly Scalable Vector Multiprocessor. SC07, November 10--16, 2007.

³ Нет необходимости в программной поддержке когерентности, как, например, в Cray BlackWidow. Но предполагается, что операционная система оптимизирована под архитектуру NUMA.

Новый тип наводок в МПП с локальной металлизацией в экранах

Тимофеев В. К.

В современных САПР МПП программы расчета искажений сигналов и перекрестных наводок в смежных линиях передачи производятся, исходя из безусловного обеспечения разработчиком основных высокочастотных свойств линий передачи, т.е. распространение сигнала по линии передачи относительно «земли».

Автор обращает внимание разработчиков высокочастотных МПП на факт изменения структуры печатной платы всеми программами САПР: замена энергетических слоев «PWR», содержащие фрагменты локальной металлизации, и слоев «GND» сплошными слоями «PLANE», т.е. - идеальными экранами.

Проектирование многослойных высокочастотных печатных плат (МПП) имеет много особенностей, которые не учитываются традиционными САПР МПП, но которые обязан учитывать разработчик. Наиболее важная из них связана со структурой платы – порядком размещения и размерами слоев «земля» и «питание».

В СВЧ технике МПП известны следующие основные типы помех в линиях связи:

- перекрестные наводки, обусловленные емкостными и взаимно индуктивными связями;
- помехи, распространяющиеся по шинам питания между слоями питания (PWR) и земли (GND) как по линиям передачи с малым значением импеданса;
- помехи, возникающие на активном сопротивлении и индуктивности вводов слоев металлизации GND и PWR.

В большинстве практических разработок САПР рассматриваются именно помехи указанных типов, проявляющиеся в конструкциях при условии идеально качественного сопряжения различных экранов. Однако в МПП малых размеров или в МПП больших размеров, имеющих в слоях PWR области локальной металлизации различного назначения, необходимые условия сопряжения экранов для некоторых линий связи оказываются трудно реализуемыми. В подобных случаях рекомендуется использовать сопрягающие конденсаторы по всей площади каждого фрагмента локальной металлизации слоя питания, над/под которыми проходят активные и пассивные связи [5, 6], что увеличивает стоимость конструкции, затрудняет размещение основных логических элементов, снижает трассировочную способность слоев

МПП [4]. Поэтому проектировщикам важно хорошо ориентироваться в различных практических ситуациях и, в частности, в зависимостях величины помехи от размеров локальной металлизации.

«Следует обратить внимание, что схемы высокого быстродействия, имеющие выделенные слои металлизации, в том числе выполняющие технологические функции выравнивания поля металлизации, описываются программами САПР весьма искусственно» [2]. Оптимальные способы трассировки линий связи для новых изделий являются пока ещё важной и до конца не решенной проблемой. Поэтому поиск требуемых конструктивных решений, выбор соответствующей топологии линий связи, включающий минимизацию паразитных наводок, несмотря на постоянное совершенствование средств САПР, продолжает оставаться длительным итерационным процессом при проектировании [2]. Тем самым подчеркивается недостаточность внимания со стороны разработчиков программного обеспечения САПР МПП анализу структурной организации МПП и связанными с ней характеристикам линий передачи.

Автор настоящей статьи для решения обозначенной проблемы использовал современную систему моделирования высокочастотных и сверхвысокочастотных (СВЧ) устройств фирмы Applied Wave Research «Microwave Office».

Суть проблемы

В современных САПР МПП программы расчета искажений сигналов и перекрестных наводок в смежных линиях передачи производятся, исходя из безусловного обеспечения разработчиком основных высокочастотных свойств линий передачи, т.е. распространение сигнала по линии передачи относительно «земли» (рис.1).

В данной работе конкретизируются физические условия возникновения наводок, оценивается амплитуда помех в линиях связи в зависимости от размеров локальных металлизаций в экранах, помехи от конструктивных параметров сопряженных и несопряженных экранов, используемых для подвода питающих напряжений с различными потенциалами.

Рассматриваемые структуры рис. 1. призваны показать существенные конструктивные различия сопряженных и несопряженных экранов.

В классической полосковой линии передачи (с сопряженными слоями GND) основная часть электромагнитной энергии сосредоточена в области между проводником и экранами и утроенной шириной сигнального проводника в поперечном направлении относительно направления распространения сигнала. По-видимому, в несопряженных экранах часть емкости сигнального проводника активной линии образует емкостный делитель с проводником пассивной линии через слой PWR, вследствие чего паразитный сигнал захватывает область, значительно превышающую утроенную ширину сигнального проводника.

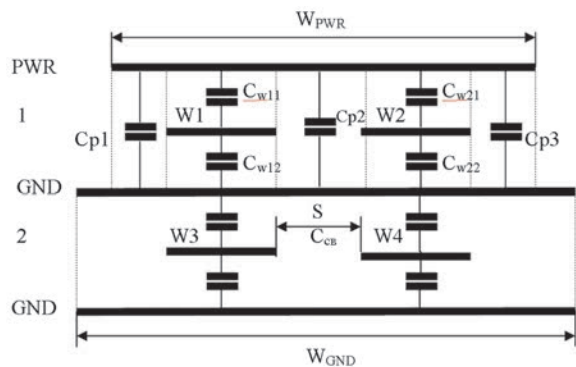


Рис. 1. Структура многослойной печатной платы, емкости плоских частей сигнальных проводников относительно слоев PWR и GND и емкости экранов в промежутках между линиями передачи.

Нижняя часть структуры – симметричная классическая полосковая линия передачи с сопряженными экранами, со структурой “GND” – “GND”. Экраны являются сопряженными, когда для высокочастотного сигнала могут взаимно

меняться местами. Непременным условием сопряжения экранов является электрическая связь пары плоскостей в непосредственной близости от входов и выходов анализируемых связанных линий передачи. Экранирующие слои могут быть связаны короткозамкнуто или с помощью сосредоточенных конденсаторов боНижняя часть структуры – симметричная классическая полосковая линия передачи с сопряженными экранами, со структурой “GND” – “GND”. Экраны являются сопряженными, когда для высокочастотного сигнала могут взаимно меняться местами. Непременным условием сопряжения экранов является электрическая связь пары плоскостей в непосредственной близости от входов и выходов анализируемых связанных линий передачи. Экранирующие слои могут быть связаны короткозамкнуто или с помощью сосредоточенных конденсаторов большой емкости. В этом случае расположения сигнальных проводников (W3 и W4) на расстоянии S, превышающем утроенную ширину сигнального проводника, перекрестной наводкой, обусловленной емкостью связи C_{cb} по зазору S, можно пренебречь.

Верхняя часть структуры МПП “PWR” – “GND” содержит сигнальные проводники, расположенные между слоем питания (PWR) и землей (GND). Экранирующие слои не могут быть связаны короткозамкнуто, поскольку потенциал слоя PWR отличен от потенциала GND. Вариант с установленными сосредоточенными емкостями большой величины известен, требует усложнения конструкции и стоимости изделий. В статье рассматривается характер помех в структуре с несопряженными экранами.

Эквивалентная схема связанных линий передачи с несопряженными экранами.

Рассмотрим эквивалентную схему связанных линий передачи структуры «1», верхняя часть рис. 1.

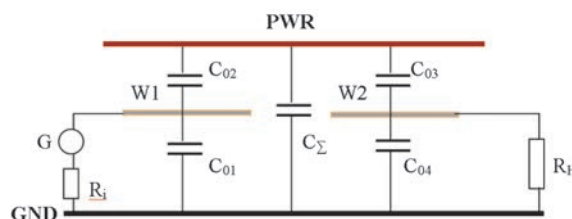


Рис. 2. Эквивалентная схема линии передачи с несопряженными экранами.

$C_{\Sigma} = C_{P1} + C_{P2} + C_{P3}$ - Сумма частичных емкостей

слоя PWR относительно слоя GND на рис.1.

$C_{01} = C_{02} = C_{03} = C_{04} = C_0$ – погонные собственные емкости плоских частей сигнальных проводников относительно экранов. Краевыми емкостями сигнальных проводников рис.2 пренебрегаем, поскольку они должны учитываться только при ширине сигнальных проводников соизмеримых с толщиной металлизации. В рассматриваемом случае $W=0.25\text{мм}$, а $t=0.035\text{мм}$, т.е. отличаются почти на порядок. Теория краевых емкостей подробно рассмотрена в [1].

В тех случаях, когда ширина локальной металлизации, под которой проходят сигнальные проводники, становится соизмеримой с шириной сигнальных проводников (при $C_{p1} = C_{p2} = 0$ и при $S = W$), можно считать, что сумма последовательно включенных емкостей C_{03} и C_{04} равна емкости C_{Σ} рис.2.

Емкостное сопротивление каждой емкостной составляющей для эквивалентной схемы рис.2. может быть оценено по формуле:

$$R_{Ci} = \frac{1}{j\omega C_i} \quad (1)$$

$$R_e = R_{C\Sigma} \parallel (R_{C03} + R_{C04}), \quad (2)$$

Где R_e – эквивалентное сопротивление параллельных ветвей.

Напряжение на локальной металлизации PWR равно

$$U_{\text{pwr}} = \frac{E \cdot R_e}{(R_i + R_{C02} + R_e)} \quad (3)$$

Где E – ЭДС генератора G.

Напряжение наводки на пассивной линии передачи или на нагрузочном сопротивлении R_n может быть определено по формуле:

$$U_n = U_{\text{pwr}} \left(\frac{R_{C\Sigma} \parallel (R_{C03} + R_{C04})}{R_{C03} + R_{C04} + R_{C\Sigma}} \right) \quad (4)$$

Для большинства генераторов импульсной техники внутреннее сопротивление генератора R_i равняется приблизительно 10 - 40 Ом. Поэтому в эквивалентной схеме рис. 2. можно пренебречь влиянием емкости C_{01} , включенной параллельно внутреннему сопротивлению генератора R_i (по формуле [1] для проводника 100мм, на частоте следования импульсов 100МГц емкостное сопротивление проводника $R_{C01} = 150$ Ом). Для упрощения качественных оценок примем $R_n \gg Z_0$. Тогда, при увеличении емкости C_{Σ} , т.е. при увеличении ширины металлизации W_{PWR} , соответствующее

емкостное сопротивление $R_{C\Sigma}$ стремится к нулю, следовательно, U_n также стремится к нулю (4).

При условии, что $C_{01} = C_{02} = C_{03} = C_{04} = C_0$ (рис.2.), когда R_i существенно меньше $2R_{C0}$,

$$U_{\text{pwr}} = E \cdot R_{C0} / (R_i + 2R_{C0}) = E \frac{1}{j\omega C} / (R_i + 2 \frac{1}{j\omega C}) = E \frac{1}{j\omega C} \cdot \frac{j\omega C}{j\omega C R_i + 2} \quad (5)$$

Очевидно, что частоты и емкости сокращаются, а напряжения на нагрузках в начале и конце линий передачи не зависят ни от частоты, ни от емкости пассивных линий передачи, следовательно, не зависит и от длины линий связи. Еще раз подчеркиваем, что указанное свойство характерно только при выполнении условия высокоомности состояний всех входов интегральных схем и малости значения внутреннего сопротивления генератора, подключенного к активной линии передачи.

Первое условие, безусловно, выполняется для всех низкочастотных составляющих спектра последовательности импульсных сигналов, следовательно, импульсная помеха должна проявляться в течение всего времени действия каждого импульса. Эта помеха не затухает с течением времени, как это свойственно перекрестным помехам в классических параллельных связанных линиях передачи, расположенных на малых расстояниях друг от друга.

Из соотношения (5) следует, что $U_{\text{pwr}} = E/2$, а напряжение помехи на нагрузке при равенстве емкостей C_{03} , C_{04} , в два раза меньше - $U_n = E/4$ т.е. при амплитуде импульсного сигнала 3300mV амплитуда сигнала помехи может составить 825mV. Итак, следует признать, что область значений помех (как функции от ширины локальной металлизации для несопряженных экранов) находится в области значений от 0 до $E/4V$.

При моделировании указанного процесса следует ожидать соответствия формы сигнала помехи форме сигнала задающего генератора и независимости амплитуд помех от длины области связи, расстояний “S” между активными и пассивными проводниками при $S > 3W_1$.

Моделирование паразитных связей

Моделирование паразитных связей в линиях передачи производилось с использованием программы "Microwave Office 2001 v.4.02".

Моделирование паразитных связей в линиях передачи с несопряженным экраном шириной 1мм. Основные электрофизические параметры линий передачи с минимальной шириной локальной металлизации:

$$W_{\text{pwr}} = W_1 + W_2 + S_1 + S_2 = 4 \cdot 0.25 \text{ мм.}$$

Максимальное значение напряжения помехи на пассивной ближайшей линии передачи 593mV. Это значение хорошо согласуется с ранее сделанной оценкой $U_1 / 4 = 2825 / 4 = 706 \text{ mV}$. Показанное отличие можно объяснить присутствием второго пассивного проводника в данном макете моделирования (в эквивалентной схеме не показан). Моделирование производилось только с целью подтверждения ранее сделанных аналитических оценок. Дополнительные материалы по результатам моделирования представлены в [3]. Общий случай и результаты анализа рассматриваются далее.

Моделирование паразитных связей в линиях передачи производилось с использованием программы "Microwave Office 2001 v.4.02".

Моделирование паразитных связей в линиях передачи с несопряженным экраном шириной 0.75мм. Подобный случай возможен при прохождении 2-х и более проводников над/под шиной питания, например, вентилятора.

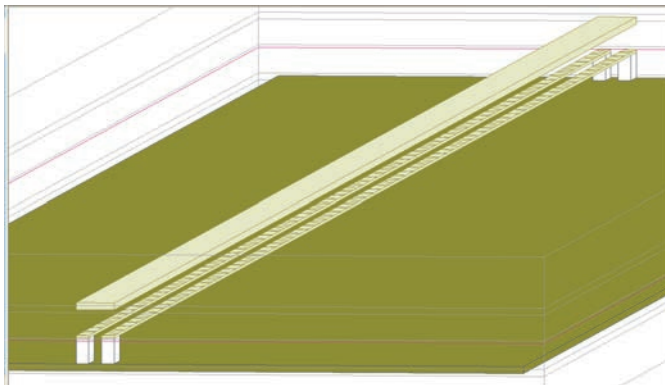
Ширина сигнальных проводников – $W = 0.25 \text{ мм}$.

Ширина локальной металлизации равна утроенной ширине сигнальных проводников – 0.75 мм .

Длина проводников и локальной металлизации – 100 мм .

Частота следования импульсных сигналов – 100 МГц .

2.1 Высокоомные нагрузки на концах линий передачи.

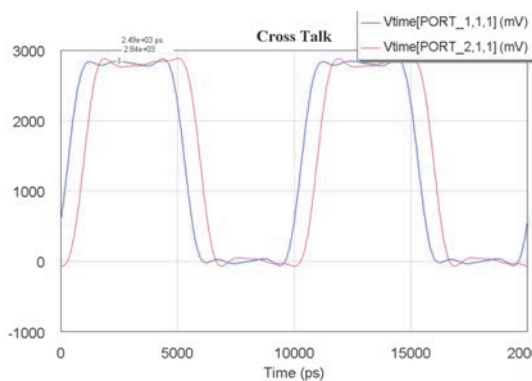


Номинальные значения портов

Variable Tuner					
Tune	P4:Z	P3:Z	P2:Z	P1:TW	P1:Z
Nom->	5300	5300	57.5	4500	10.0016
Max->	10600	10600	115	9000	20.0032
Close	-	-	-	-	-
Save	-	-	-	-	-
Reset	-	-	-	-	-
Revert	-	-	-	-	-
Sweep	-	-	-	-	-
Freeze	-	-	-	-	-
Clear	-	-	-	-	-
Help	-	-	-	-	-
Min->	2650	2650	28.75	2250	5.0008

Рис. 3. Объемное представление линий связи с узкой локальной металлизацией над ними.

Форма и амплитуда сигнала генератора - агрессора. $U_g = 2840 \text{ mV}$



Формы и амплитуды сигналов на линии - жертве. Положительный выброс $U_{p1} = 347 \text{ mV}$.

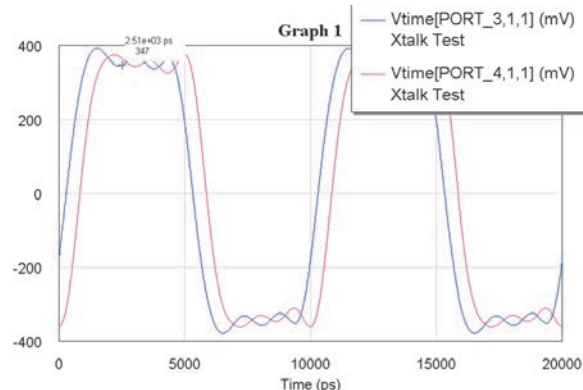


Рис. 4. Формы и амплитуды сигналов генератора и сигналов на линии - жертве.

Ранее отмечалось, что при ширине локальной металлизации, равной утроенной ширине сигнального проводника, помеха должна равняться 1/4 амплитуды сигнала генератора.
 $U_g/4 = U_{p1} + U_{p2}$; $U_g = 2840/4 = 710$.

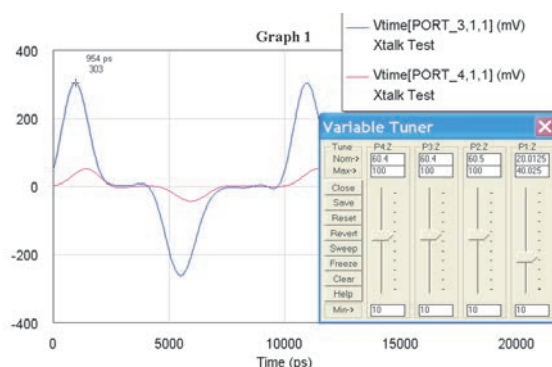
Результаты моделирования:

$U_{p1} + U_{p2} = 347\text{mV} + 331\text{mV} = 678\text{mV}$ – отклонение от ожидаемой величины - 4.5%.

Подобное отклонение вполне допустимо в связи с тем, что ранее было сделано допущение о равенстве нулю перекрестной связи между торцами смежных линий, которые расположены на расстоянии S меньшем утроенной ширины сигнального проводника W. В рассматриваемом случае $S = W$.

2.2 Низкоомные нагрузки на концах линий передачи.

$R_{p3} = R_{p4} = 60 \text{ Ohm}$



$R_{p3} = 20 \text{ Ohm}; R_{p4} = 60 \text{ Ohm}$

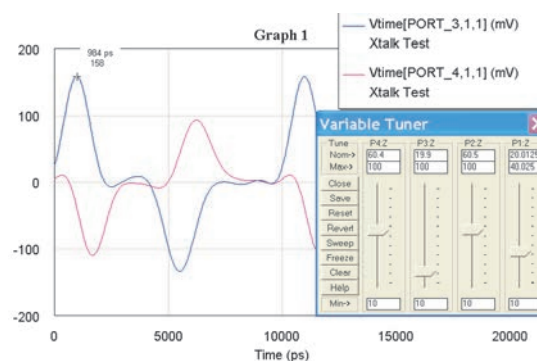


Рис. 5. Формы и амплитуды сигналов генератора и сигналов на линии – жертве.

При $R_{p3} = R_{p4} = 60 \text{ Ohm}$ амплитуда помехи $U_p = 303 \text{ mV}$ соизмерима с уровнем помехи $U_p = 347 \text{ mV}$ в варианте высокоомных нагрузок на концах линий передачи.

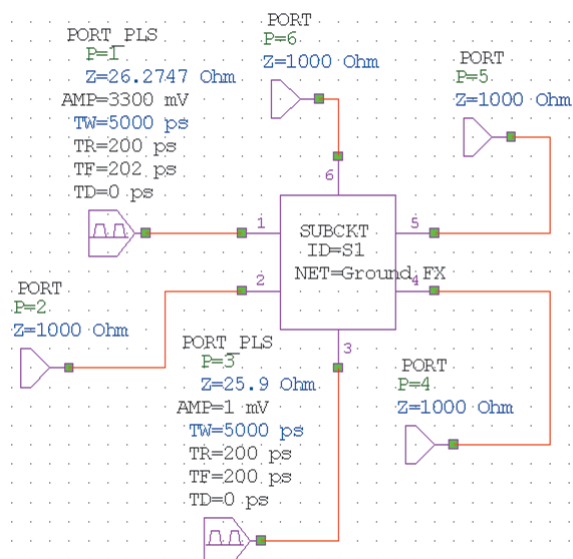
В типовом варианте $R_{p3} = 20 \text{ Ohm}; R_{p4} = 60 \text{ Ohm}$ помеха снижается $U_p = 158 \text{ mV}$ из-за уменьшения сопротивления порта P3 до 20 Ohm.

Моделирование паразитных связей в линиях передачи с несопряженным экраном шириной 8мм.

На рис. 6 представлена общая электрическая схема включения портов и связанных линий передачи.

Данные импульсного сигнала соответствуют активному порту P1.

Порт P3 для данного варианта был заменен на пассивный порт с сопротивлением $R=1000 \text{ Ohm}$.



- P1 - активный порт, расположенный в начале активной линии передачи.
- P2 - пассивный порт в конце активной линии передачи
- P3, P5 - пассивные порты, расположенные в началах пассивных линий передачи
- P4, P6 - пассивные порты, расположенные в конце пассивных линий передачи
- F - частота следования импульсов..... 100МГц
- Tw - длительность импульса 5нс
- Tr - длительность роста фронта импульсного сигнала 0.2нс
- Tf - длительность спада фронта импульсного гнала 0.2нс

Рис. 6. Схема подключения портов к плате с сигнальными проводниками.

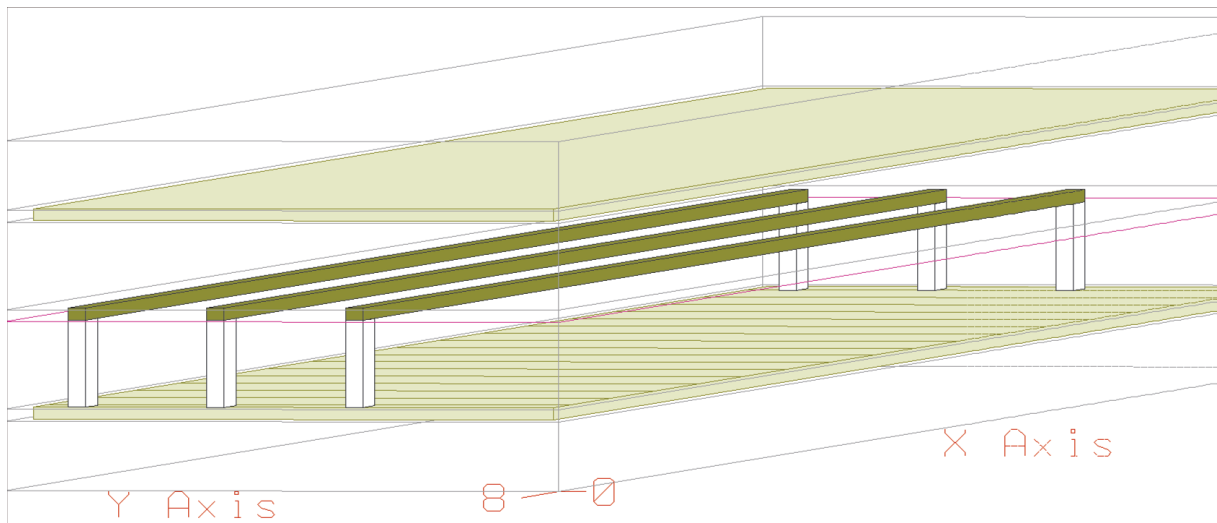


Рис. 7. Линии передачи, расположенные между экранами.

На рис.7. представлена объемная структура высокочастотных линий передачи.

Для наглядности изображение выполнено с большим коэффициентом увеличения. Изменение масштаба изображения не влияет на электрофизические параметры линий передачи.

В качестве генератора использован точечный порт, включенный одним концом на землю, другим – на линию передачи.

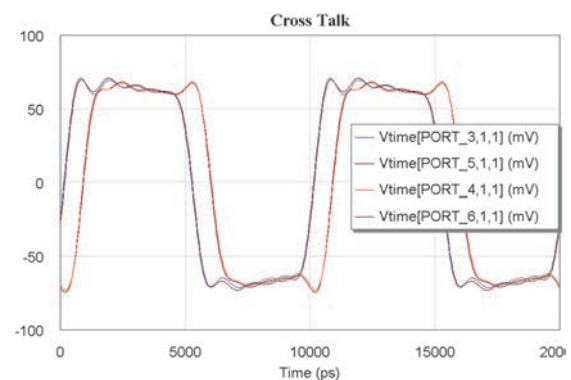
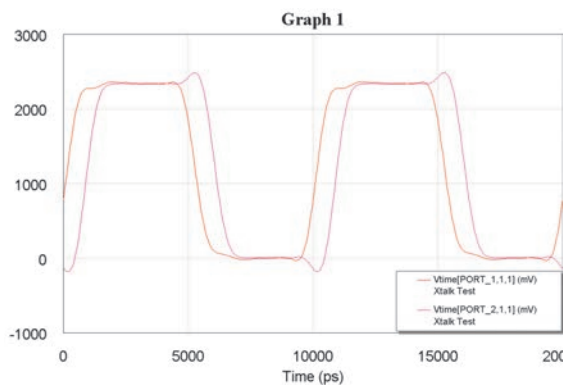
Амплитуда (ЕДС) Генератора	3300 mV.
Ri генератора (порт P1)	20 Ом.
L1, L2, L3 – длины исследуемых линий передачи	128мм
W1=W2=W3=W – ширина линий передачи	0.25мм
- ширины слоев «земля» и несопряженного экрана W_{GND}	8 и 7.5мм
t - толщина металлизации в слоях сигнал и «земля»	0.035мм
ϵ_r – относительная диэлектрическая проницаемость диэлектрика между сигнальным слоем и слоями «земля» и PWR	4.7
T_e – толщины диэлектрических слоев	0.25мм
S1 – расстояние от края активной до края первой пассивной линии передачи, расположенной в верхней части окна “Ground FX:1”	1.75мм
S2 – расстояние до второго пассивного проводника	3,75мм
Значения проводимости металлизированных медных слоев	$5.88e^{+07} S/m$,
т.е. $R_{dc} = 0.00048$ Ом на квадрат.	

3.1 Моделирование паразитных связей в линиях передачи.

Генератор $R_i=20$ Ohm;
нагрузка $R=50$ Ohm

Жертвы: $R_n = 3k$ Ohm

Формы сигналов совпадают с сигналом генератора.



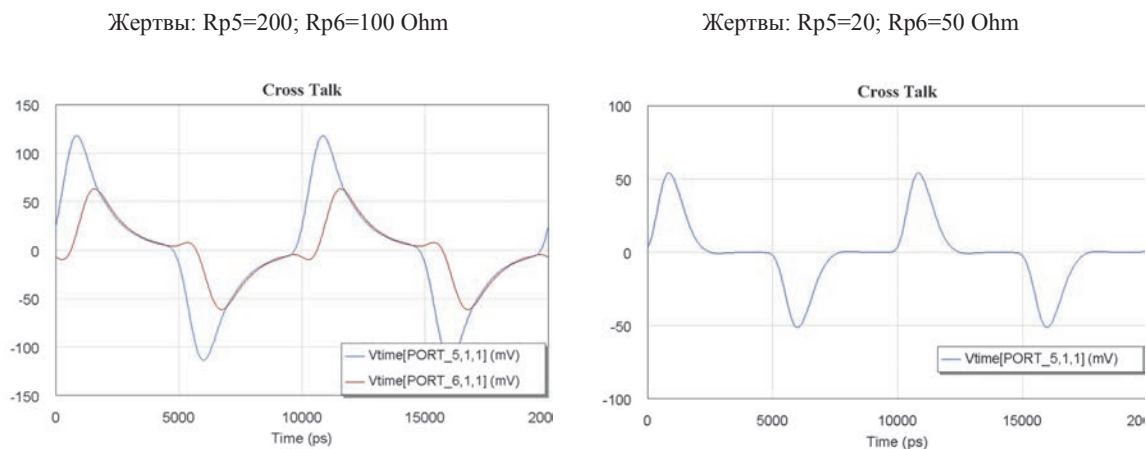


Рис. 8. Формы сигналов помех в линиях передачи, содержащих нагрузки в низкоомных и высокоомных состояниях.

Напряжения помех на высокоомных нагрузках, установленных в начале и конце линий передачи, не зависят ни от частот спектра импульсного сигнала, ни от емкости пассивных линий передачи, следовательно, не зависят и от длины линий связи. Важно отметить, что значение помехи на второй пассивной линии передачи почти не отличается от уровня помехи в первой пассивной линии и то, что формы сигналов помех на высокоомных портах ($R_n=3\text{ kOhm}$) мало отличаются друг от друга и практически совпадают с формой сигнала генератора (см. рис. 8., «Жертвы: $R_n = 3\text{ k Ohm}$ »).

Указанное свойство возможно только при условии совпадения частотного спектра импульсного сигнала помехи со спектром сигнала генератора.

Паразитный сигнал в линиях передачи, содержащих низкоомную нагрузку

Во многих случаях линии связи, на которые наводится паразитный сигнал, содержат некоторое множество портов (нагрузок), один из которых находится в низкоомном состоянии.

Значения сопротивлений портов в рассматриваемом случае следующие:

$R_1=20\text{ Ом}$; $R_2=50\text{ Ом}$; $R_5=200\text{ Ом}$; $R_6=100\text{ Ом}$.

Порт R_3 (рис.9) для этого варианта является активным портом, но амплитуда импульсного сигнала выбрана равной 1 mV . Предложенный прием позволяет при сохранении всех внутренних параметров генератора (для сигнала помехи он является нагрузкой) на нем полностью выделить сигнал помехи. В противном случае помеху в несколько десятков mV сложно было бы анализировать на фоне импульса с амплитудой в 3300 mV . Формы сигналов имеют характерные искажения на линиях-жертвах, обусловленные влиянием сопротивлений низкоомных нагрузок на емкостные делители. Ранее для формулы (5) делалось допущение: $R_n \gg Z_0$.

На рис. 8., в окне «Жертвы: $R_5=20$; $R_6=50\text{ Ohm}$ » представлены формы сигналов для линий связи, согласованных на конце. В классических полосковых смежных линиях передачи, расположенных на больших расстояниях друг от друга (3.75 мм), амплитуда помехи на два порядка меньше.

Положительные и отрицательные выбросы напряжений симметричны относительно нуля подтверждают емкостный характер помех.

Выводы

- Рассмотрен тип перекрестных помех в смежных линиях передачи, связанных с отсутствием в «САПР МПП» требования по безусловному сопряжению экранов.
- Подчеркивается необходимость сопряжения экранов на выходе ИС и в конце линии передачи на входе другой ИС.
- Сопряжение экранов с различными потенциалами, в принципе, может осуществляться с помощью конденсаторов, однако следует признать этот метод малоэффективным.

- В МПП больших размеров появляются дополнительные перекрестные наводки в области локальной (узкой) металлизации различного назначения, под/над которыми проходят активные и пассивные связи.
- В заданных размерах МПП величины наводок этого типа на расстояниях, превышающих 3-х кратную ширину сигнального проводника, практически не зависят от расстояния между активной и пассивными линиями передачи.
- Импульсная помеха проявляется в течение всего времени действия каждого импульса.
- Напряжения помех на высокоомных нагрузках, установленных в начале и конце линий передачи, не зависят ни от частот спектра импульсного сигнала, ни от емкости пассивных линий передачи, следовательно, не зависят и от длины линий связи.
- Помеха не затухает с течением времени действия импульса, как это свойственно перекрестным помехам в классических параллельных связанных линиях передачи.
- Линии передачи, содержащие хотя бы по одной низкоомной нагрузке, подвержены воздействию помех, которые не повторяют форму сигнала генератора.

Рекомендация разработчикам МПП

Автор обращает внимание разработчиков высокочастотных МПП на факт изменения структуры печатной платы всеми программами САПР: замена энергетических слоев "PWR", содержащие фрагменты локальной металлизации и слоев "GND" сплошными слоями "PLANE", т.е. - идеальными экранами.

Рекомендуется проектировать платы с сопряженными экранами. Результаты моделирования показывают, что пренебрежение сопряжением экранов в самых неблагоприятных случаях (высокоомные состояния нагрузок на концах линий-жертв) опасно появлением неучтенных помех от нескольких десятков милливольт до 0.25 амплитуды импульсного сигнала с формой сигнала помехи, совпадающей с формой сигнала генератора.

Рекомендация разработчикам ПО «САПР МПП»

Автор обращает внимание разработчиков САПР на недопустимость замены (без согласия разработчиков МПП) энергетических слоев "PWR" и "GND", содержащих фрагменты локальной металлизации, сплошными слоями "PLANE", т.е. - идеальными экранами.

Расчет перекрестных наводок программами анализа искажений сигналов должен производиться с учетом дополнительных помех, обусловленных наличием локальной металлизации, расположенной над/под группой отрезков линий передачи.

Литература

1. А. Л. Фельдштейн, Л. Р. Явич. Справочник по элементам волноводной техники, «Советское радио», Москва, 1967г.
2. В. Разевиг, «Анализ паразитных эффектов печатных плат». Electronic Design Automation (EDA)—express. 2000. № 2. С. 22–23.
3. В. К. Тимофеев, В. В. Лозовой. Анализ искажений сигналов МПП СВЧ диапазона, «ЕДА EXPRESS», № 6, 2002г. С. 29-34
4. В. К. Тимофеев. Перекрестные наводки в ПП, обусловленные применением энергетических слоев в качестве экранов высокочастотных линий передачи, «ЕДА EXPRESS», № 8, 2003г. С. 15–18.
5. Power Ground-Reference Plane Decoupling Analysis of Design Alternatives Bruce Archambeault, Ph.D. Research Triangle Park, NC Barch@us.ibm.com
6. Chen, M.Xu, T. Hubing, J. Drewniak, T. Van Doren, and R. DuBroff, "Experimental evaluation of power bus decoupling on a 4-layer printed circuit board," Proc. of the 2000 IEEE International Symposium on Electromagnetic Compatibility, Washington D.C., August

Открытие законов натурального ряда и перспективы их применения

Хренов В. П.

Заместитель директора Института систем и технологий безопасности РосНОУ.

*В последовательности простых чисел
есть тайна, непостижимая человеком.*

Л. Эйлер

Заданный объём доклада вынуждает ограничиться формулировками законов и их доказательной базой.

Основным препятствием постижению закономерности образования простых чисел (ПЧ) послужило отсутствие понимания качественных отличий между самими ПЧ. Прежде всего, удалось понять, что множество ПЧ состоит из трёх качественно отличных подмножеств.

Доказательство столь обязывающих утверждений начнём с аксиомы 1:

Два первых ПЧ 2 и 3 образуются последовательным присоединением 1 к 1 по основной формуле натурального ряда $N = 1 + 1 \cdot n$ при $n \leq 2$, то есть ${}_1P = \{2, 3\}$, которые назовём *фундаментальными* и отметим их **зелёным** и **оранжевым** цветом.

Такой способ образования ПЧ на числе 4 прекращается и больше не возобновляется. Они необходимы для образования двух самых мощных множеств: **чётных** чисел (кратных 2) и чисел кратных 3, которые по аналогии следует называть *нечётными*. Все остальные простые числа образуются иначе. Как они образуются, показывает теорема 1:

«Все простые P (за исключением 2 и 3) и все составные C числа (за исключением кратных 2 и 3) принадлежат множествам $PUC = 6n - 1 = -1 \pmod 6$; $PUC = 6n + 1 = 1 \pmod 6$ ».

Числа принадлежащие множеству $6n - 1$ (с 1 недостающей для делимости на 6), назовём *отрицательными* и отметим - синим цветом.

Числа принадлежащие множеству $6n + 1$ (с 1 избыточной для делимости на 6), назовём *положительными* и отметим - красным цветом.

Доказательство. Возьмём любое ПЧ (кроме 2 и 3). Слева (на 1 меньше) и справа (на 1 больше) от него могут быть только чётные числа. Всего получается три числа подряд, значит одно из чётных чисел (слева или справа) одновременно кратно 3, то есть равно $6n$. Это значит, что любое ПЧ принадлежит либо множеству $P = 6n - 1 = -1 \pmod 6$, либо $P = 6n + 1 = 1 \pmod 6$.

Таким образом, первая часть теоремы доказана.

Теперь докажем, что все *составные* числа (СЧ с делителями ≥ 5) так же принадлежат множествам $6n - 1 = -1 \pmod 6$ и $6n + 1 = 1 \pmod 6$.

С образованием числа 6 натуральный ряд обретает цикличность, которая позволяет «нарезать» весь натуральный ряд на элементарные циклы - интервалы от $6n$ до $6(n+1)$.

от $6n$ до $6(n+1)$
..., 24, 25, 26, 27, 28, 29, 30, ...
..., 30, 31, 32, 33, 34, 35, 36, ...
..., 36, 37, 38, 39, 40, 41, 42, ...

Все **чётные** числа вида ($6n, 6n+2, 6n+4$ и $6(n+1)$) не могут быть СЧ. Числа вида $6n+3$ так же не могут быть СЧ, так как принадлежат к множеству *нечётных* (кратных 3) чисел. Для СЧ остаются только пози-

¹ Вновь вводимые понятия и термины будут выделены полужирным курсивом.

ции чисел вида $6n+1$ и $6n-1$. Как было доказано, все ПЧ имеют вид либо $6m-1$, либо $6m+1$, а их произведения:

$$(6m-1) \cdot (6m-1) = 36m - 2 \cdot 6m + 1 = 6(6m-2m) + 1 = \underline{6n+1}, \text{ где } n = (6m-2m);$$

$$(6m-1) \cdot (6m+1) = 36m - 1 = 6 \cdot 6m - 1 = \underline{6n-1}, \text{ где } n = 6m;$$

$$(6m+1) \cdot (6m+1) = 36m + 2 \cdot 6m + 1 = 6(6m+2m) + 1 = \underline{6n+1}, \text{ где } n = (6m+2m)$$

всегда приводятся к виду либо $6n-1$, либо $6n+1$.

Таким образом, все ПЧ (за исключением 2 и 3) и все СЧ (за исключением кратных 2 и 3) принадлежат множествам $P \cup C = 6n-1 = -1 \pmod 6$ и $P \cup C = 6n+1 = 1 \pmod 6$. ■

На основании аксиомы 1 и теоремы 1 можно вывести лемму 1.

«Множество всех первых чисел (ПЧ) состоит из трёх качественно различных подмножеств:

- *фундаментальных* ${}_1P = 1+1 \cdot n = \{2, 3\}$;
- *отрицательных* $P = -1 \pmod 6$;
- *положительных* $P = 1 \pmod 6$.

$$P = {}_1P \cup P \cup P = 2, 3, \quad 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, \dots$$

Любое СЧ в соответствии с основной теоремой арифметики имеет единственное разложение на множители. На вопрос: «К какому множеству *отрицательных* или *положительных* СЧ принадлежит конкретное СЧ в зависимости от количества и качества, образующих его множителей?» отвечает **теорема 2:**

«Произведение нескольких *отрицательных* p_i и *положительных* p_i ПЧ дает *положительное СЧ* c_i при четном количестве *отрицательных* p_i и *отрицательное СЧ* c_i - при ином количестве p_i ».

Примеры:

$$p_i \cdot p_i = c_i; \quad p_i \cdot p_i = c_i; \quad p_i \cdot p_i = c_i$$

$$5 \cdot 5 = 25(6 \cdot 4 + 1); \quad 5 \cdot 7 = 35(6 \cdot 6 - 1); \quad 7 \cdot 7 = 49(6 \cdot 8 + 1)$$

Доказательство. Возьмём любое СЧ с любым количеством *отрицательных* p_i и *положительных* p_i . Группируя их попарно можно получить СЧ либо вида $6n-1$, либо $6n+1$, как это было доказано путём преобразования их произведений в теореме 1. На последнем этапе по парных преобразований получаются три сочетания: $p_i \cdot p_i = c_i$; $p_i \cdot p_i = c_i$; $p_i \cdot p_i = c_i$, что доказывает справедливость теоремы 2. ■

Следующим шагом постижения законов натурального ряда стало открытие нового математического понятия p – *аддитивных* прогрессий, то есть арифметических прогрессий, в постоянной и переменной частях которых имеется только одно ПЧ p_i .

По необъяснимым причинам математики прошли мимо **аксиомы 2:**

«Последовательное присоединение к простому числу p_i одного и того же кратного количества этого же p_i образует арифметическую прогрессию с наименьшим общим множителем $НОМ = p_i$.

$$\{p_i, (p_i + k \cdot p_i), (p_i + 2 \cdot k \cdot p_i), (p_i + 3 \cdot k \cdot p_i), \dots\} = \{p_i + p_i \cdot k \cdot n\} = p_i \{1 + k \cdot n\}$$

Если к p_i последовательно прибавлять кратные количества этого же p_i , то образуется арифметическая прогрессия с *наименьшим общим множителем* (так же новый математический объект в последующем обозначаемый *НОМ*), которая путём сложения позволит получать последовательность произведений ПЧ, причём это p_i является *НОМ*. Открытые понятия и их свойства позволяют сформулировать «**Закон p – аддитивных прогрессий**», по которому образуются *фундаментальные* ПЧ и все СЧ натурального ряда: «Множество всех простых чисел образуют множество p – *аддитивных* прогрессий по формулам

$$p_i C = p_i \cdot k_i + p_i \cdot k_j \cdot n,$$

где $n = 0, 1, 2, \dots$; k_i и k_j – константы, принимающие следующие значения:

- у *фундаментальных* ПЧ при $p_0 = 1, k_i = 1$ и $k_j = 1$ ${}_1P = 1 + 1n = \{2, 3\}$;
- у *чётных* чисел при $p_1 = 2, k_i = 1$ и $k_j = 1$ ${}_2C = 2 + 2n = 4, 6, 8, 10, \dots$;
- у *нечётных* чисел при $p_2 = 3, k_i = 1$ и $k_j = 2$ ${}_3C = 3 + 3 \cdot 2n = 9, 15, 21, \dots$;
- у СЧ $\sum p_i C$ с *НОМ* = $p_i \rightarrow k_i = p_i$ или с *НОМ* = $p_i \rightarrow k_i = p_{i+1}$, а $k_j = 6$;

$$\sum p_i C = \{p_1 \cdot p_1 + p_1 \cdot 6m\} \cup \{p_1 \cdot p_2 + p_1 \cdot 6m\} \cup \dots \cup \{p_i \cdot p_i + p_i \cdot 6m\} \cup \{p_i \cdot p_{i+1} + p_i \cdot 6m\} \cup \dots$$

где \cup – знак объединения множеств, i – индексы ПЧ $= \{1, 2, 3, \dots\}$, т. е. $p_1 = 5, p_1 = 7, p_2 = 11, p_2 = 13, \dots$, а $m = \{0, 1, 2, 3, \dots\}$.

Пример: $C = \{5 \cdot 7 + 5 \cdot 6m\} \cup \{7 \cdot 11 + 7 \cdot 6m\} \cup \{11 \cdot 13 + 11 \cdot 6m\} \cup \{13 \cdot 17 + 13 \cdot 6m\} \cup \dots =$
 $= \{35, 65, 95, \dots\} \cup \{77, 119, 161, \dots\} \cup \{143, 209, 275, \dots\} \cup \{221, 299, 377, \dots\} \cup \dots;$

- у СЧ $\sum p_i C$ с $HOM = p_i \rightarrow k_i = p_i$ или с $HOM = p_i \rightarrow k_i = p_i$, а $k_j = 6$.

$$\sum p_i C = \{p_i^2 + p_i \cdot 6m\} \cup \{p_i^2 + p_i \cdot 6m\} \cup \dots \cup \{p_i^2 + p_i \cdot 6m\} \cup \{p_i^2 + p_i \cdot 6m\} \cup \dots$$

где $i = 1, 2, 3, \dots$ - индексы ПЧ, т. е. $p_1 = 5, p_2 = 7, p_3 = 11, p_4 = 13, \dots$, а $m = 0, 1, 2, 3, \dots$

Пример: $C = \{5 \cdot 5 + 5 \cdot 6m\} \cup \{7 \cdot 7 + 7 \cdot 6m\} \cup \{11 \cdot 11 + 11 \cdot 6m\} \cup \{13 \cdot 13 + 13 \cdot 6m\} \cup \dots =$
 $= \{25, 55, 85, \dots\} \cup \{49, 91, 133, \dots\} \cup \{121, 187, 253, \dots\} \cup \{169, 247, 325, \dots\} \cup \dots$

Ограниченный объём доклада не позволяет привести полное доказательство Закона p – аддитивных прогрессий. С ним можно ознакомиться по другим публикациям [1 - 4].

Теперь, когда мы знаем формулы образования всех СЧ, просто составить формулу, по которой можно только операциями сложения и сравнения вычислять все ПЧ подряд. Для этого, памятуя теорему 1, необходимо из множества $6n - 1$ исключить (\setminus - знак исключения множеств) подмножество всех *отрицательных* СЧ $\sum p_i C$, а из множества $6n + 1$ исключить подмножество всех *положительных* СЧ $\sum p_i C$ и получить *формулы отрицательных* и *положительных* простых чисел:

$$P = \{6n-1\} \setminus \sum p_i C = \{6n-1\} \setminus [\{p_1 \cdot p_1 + p_1 \cdot 6m\} \cup \{p_1 \cdot p_2 + p_1 \cdot 6m\} \cup \{p_2 \cdot p_2 + p_2 \cdot 6m\} \cup \dots \\ \cup \{p_i \cdot p_i + p_i \cdot 6m\} \cup \{p_i \cdot p_{i+1} + p_i \cdot 6m\} \cup \{p_{i+1} \cdot p_{i+1} + p_{i+1} \cdot 6m\} \cup \dots]$$

$$P = \{6n+1\} \setminus \sum p_i C = \{6n+1\} \setminus [\{p_1 \cdot p_1 + p_1 \cdot 6m\} \cup \{p_1 \cdot p_2 + p_1 \cdot 6m\} \cup \{p_2 \cdot p_2 + p_2 \cdot 6m\} \cup \dots \\ \dots \cup \{p_i \cdot p_i + p_i \cdot 6m\} \cup \{p_i \cdot p_i + p_i \cdot 6m\} \cup \{p_{i+1} \cdot p_{i+1} + p_{i+1} \cdot 6m\} \cup \dots]$$

и формулировку Закона простых чисел: «Множество всех простых чисел состоит из подмножества фундаментальных ПЧ ${}_1P$, подмножества *отрицательных* ПЧ P равного множеству $6n-1$ за исключением совокупности всех p - аддитивных прогрессий *отрицательных* СЧ $\sum p_i C$ и подмножества всех *положительных* ПЧ P равного множеству $6n+1$ за исключением совокупности всех p - аддитивных прогрессий *положительных* СЧ $\sum p_i C$ ».

Теперь мы знаем законы образования всех чисел натурального ряда и можем сформулировать «Основной закон арифметики», расщепляющий натуральный ряд на 7 однозначно идентифицируемых (цветом и наклоном шрифта) по их качеству подмножеств чисел: «Натуральный ряд N состоит из $0; 1$; двух фундаментальных простых чисел ${}_1P = 2, 3$; подмножеств *отрицательных* P и *положительных* P простых чисел; p - аддитивных прогрессий чётных чисел ${}_2C$, нечётных чисел ${}_3C$, подмножеств *отрицательных* $\sum p_i C$ и *положительных* $\sum p_i C$ составных чисел».

$$N = 0, 1 \cup P \cup C = 0, 1, {}_1P \cup P \cup P \cup {}_2C \cup {}_3C \cup \sum p_i C \cup \sum p_i C =$$

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, \dots$$

Так математика получила цветное – качественное зрение.

Этот закон однозначно расщепляет натуральный ряд на 7 качественно различных множеств подобно призме Френеля, расщепляющей белый свет на 7 цветов радуги, и даёт новую, совершенно однозначную классификацию натуральных чисел.

Для сравнения покажем существующую классификацию, используя аналогичный цвет и наклон шрифта: чётные числа – *зелёным* цветом, нечётные числа – чёрным цветом, простые числа прямым шрифтом, а составные числа – *курсивом*:

61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89,

Достаточно одного взгляда, чтобы увидеть ущербность существующей пока классификации без учёта качества чисел.

Теперь уместно прокомментировать правомерность применения термина *закон** по отношению к открытым математическим закономерностям.

* - *закон* –необходимое, существенное, повторяющееся соотношение между явлениями. *З.* выражает связь между предметами, составными элементами данного предмета, между свойствами вещей, а так же свойствами внутри вещи. *З.* могут быть менее общими, более общими и всеобщими. Отражённые в мышлении *З.* составляют ядро науки. Власть человека над окружающим миром измеряется объёмом и глубиной знания его *З.* (БСЭ).

Для данных в официальном определении слова ЗАКОН понятий в нашем случае имеют место быть следующие смысловые эквиваленты: «предметами» - числовая система природы или качественно одназначная её часть (например, множество всех ПЧ); «составными элементами данного предмета» - качественно различные, однозначно идентифицируемые классы последовательностей; «свойствами вещей» – качественные свойства этих последовательностей и «свойствами внутри вещи» – вскрытая недостаточность 1-цы и избыточность 1-цы для делимости *отрицательных* и *положительных* ПЧ и СЧ без обозначения знаков арифметических действий перед этими числами.

Таким образом, доказанные выше законы в математической форме отображают **необходимое существование, устойчивое, повторяющееся отношение между предметом – натуральным рядом N , составными элементами ${}_1P, P, P, {}_2C, {}_3C, C, C$ предмета, между свойствами вещей – сочетание в P фундаментальных ${}_1P$, отрицательных P и положительных P , а так же между свойствами внутри вещи – свойство отрицательного числа P без знака вычитания перед ним и свойство положительно числа P без знака сложения перед ним, что позволяет отнести открытые законы к категории всеобщих законов.**

Ещё много копий будет сломано по поводу правомерности применения понятия «закон» к открытым закономерностям натурального ряда. Закономерностями принято называть качественно выявленные причинно-следственные связи. Когда причинно следственные связи выявлены качественно и описаны однозначно количественно – они должны иметь название Закон, тем более, что законы натурального ряда отображают фундаментальные философские законы, а после принятия природных мер пространства, времени и энергии ими можно отображать законы взаимодействий Природы. Таково видение автора. Математическое сообщество вольно называть представленные здесь открытия как угодно, что не умаляет значимость открытия, как для теории чисел, так и для всей математики, как универсального языка познания.

Кроме общенаучного значения, открытые законы имеют особое значение в области создания систем защиты информации (СЗИ). Именно здесь сугубо математическая проблема факторизации (определение делителей составного числа) была положена в основу асимметричных систем защиты информации, пригодной для широкого пользования [9].

Открытые математические закономерности позволяют создать СЗИ нового поколения на одноразовых ключах и одноразовых непериодических гаммах псевдослучайных чисел на конечных автоматах, что до настоящего времени считалось невозможным [10]. Такие СЗИ будут обладать теоретически максимальной надёжностью и быстродействием режима on line.

Началась реализация открытых законов в практическом плане – автором доклада получено свидетельство № 2005613012 от 22.09.2005 г. о регистрации программы «Линейный генератор простых чисел подряд» [2], которое получило высокую оценку коммерческой значимости. Члены Государственной комиссии в своём заключении от 26.04.2005 г. [1] подтвердили, что зависимость времени вычислений последовательностей простых чисел подряд от разрядности задаваемого диапазона вычислений носит линейный характер, что было бы невозможно без знания закона их формирования.

Новый математический базис и основанные на нём технические решения в области защиты информации, защищённые патентами [11, 12], являются первой предпосылкой для создания систем защиты информации нового поколения.

Второй предпосылкой является наличие в составе авторского коллектива разработки создателей российских дистрибутивов ALT Linux Master, на базе которого подготовлена ЗИС «Утёс – 5», сертифицированная Гостехкомиссией при Президенте РФ на 5 класс СТВ и 4 класс НДВ, а так же специальной версии дистрибутива «ИВК – Кольчуга», сертифицированной ФСТЭК и Министерством обороны на 2 класс МЭ, что позволяет использовать её для работы с грифом «сов. секретно».

Система представляет собой межсетевой экран с дополнительными функциями офисного сервера. При этом выполнение типовых задач системного администрирования максимально упрощено, что позволяет с ней работать системным администраторам, не имеющим серьёзной подготовки в области Linux.

Благодаря изложенному, Россия получила уникальный шанс на «Создание сверх защищённой распределенной операционной системы (СРОС) обеспечивающей информационную безопасность с помощью одноразовых ключей и одноразовых непериодических гамм». Создание такой системы позволит исключить в реальной зоне риска отказы в обслуживании, порчу данных в информационных системах, обеспечить надежное дублирование хранимых данных, значительно расширить доступность и увеличить быстродействие.

Эта новая информационная технология строится на базе свободного программного обеспечения (СПО) – (исходные тексты доступны разработчикам), позволяя в максимальной степени учитывать требования заказчика.

Создание СРОС даёт следующие конкурентные преимущества:

- существенно сократить расходы на легализацию программного обеспечения, используемого в России, как в госсекторе, так и в коммерческом секторе (по сравнению со стоимостью продуктов фирмы Microsoft использование СПО даёт экономию от 50 до 70 %);
- активизировать развитие отечественного производителя СПО и инвесторов, выйдя на международный рынок в виде поставщика интеллектуальных продуктов и решений, тем самым минимизировав технологическую зависимость России в области ПО;
- обеспечить сверхвысокий уровень информационной безопасности за счёт доступа к исходному коду и его независимого аудита, а также за счёт принципиально новой патентованной [2, 11, 12] технологии защиты информации, основанной на одноразовых ключах и одноразовых непериодических гаммах, генерируемых в автоматическом режиме с использованием закона формирования простых чисел;
- повысить быстродействие блоков защиты информации до режима on line;

Кроме того, открытые законы позволят получить значительный коммерческий эффект в широком спектре применений:

- Система борьбы с контрафактной продукцией и поддельными денежными знаками,
- Система дистанционной идентификации и борьбы с угонами автотранспорта,
- Система борьбы с распространением компьютерных вирусов,
- Компьютеры нового поколения на нелинейной системе счисления природы;
- Математико-биологическое обоснование теории гармонии восприятий;
- Математический аппарат для нано - технологий.

Имеются макетный образец системы, алгоритмы и схемные решения.

Аналогичных решений в области разработки защищенных операционных систем на отечественном и зарубежном рынке нет.

Потребность в разработке: банковская сфера, военно-промышленный комплекс, государственные и коммерческие организации, обычные пользователи.

Литература

1. Заключение Государственной экспертизы № 167 от 26.04.2005г (ФГУ НИИ РИНКЦЭ МИНОБРНАУКИ РОССИИ).
2. Свидетельство № 2005613012 от 22.09.2005 г. об официальной регистрации программы ЭВМ «Линейный генератор простых чисел подряд (Р 1.1)».
3. В. П. Хренов. «Проблемы и перспектива создания систем защиты информации нового поколения», журнал «Глобальная безопасность», № 3, 2005 г.
4. В. П. Хренов. «Новый этап развития систем защиты информации», журнал «Наука и технологии в промышленности», №3, 2005 г.
5. В. П. Хренов. Prime Numbers Technology (PNT)[™] – основа создания систем защиты информации (СЗИ) нового поколения и перспективы её применения в различных сегментах информационных технологий, журнал «Бизнес и Безопасность в России» №46, январь 2007 г.
6. <http://narod.yandex.ru/Начала теории натуральных чисел полн.doc.html>
7. В. П. Хренов. Начала теории натуральных чисел. Доклад на международной научной конференции «Цивилизация знаний: инновационный переход к обществу высоких технологий», 25.04.2008 г., в Российском новом университете (РосНОУ), опубликован в сборнике докладов.
8. В. П. Хренов, доклады «Исторические аспекты кризиса математики и, соответственно, естествознания» и «Новый взгляд на теорию чисел» на международной научной конференции «Образование, наука и экономика в вузах. Интеграция в международное образовательное пространство», г. Польша, 9-14 сентября 2008 г., опубликованы в сборнике докладов.
9. W. Diffie and M.E. Hellman, “New Directions in Cryptography”, IEEE Transactions on Information Theory, v. IT-22, n. 6, Nov 1977, pp. 74-84.
10. В. Schneier, Applied Cryptography, John Wiley & Sons, Inc., 1996.
11. В. П. Хренов, «Система защиты информации», патент **RU 2 325 695 C2 G06F 21/20** от 19.09.2005 г.
12. В. П. Хренов, «Способ защиты информации», патент **RU 2 325 766 C2 H04L 9/18** от 16.09.2005 г.

Современные методы функциональной верификации цифровых проектов

Шабанов Б. М., *к.т.н., доцент, первый зам. директора ГУ МСЦ*

Яицков А. С., *д.т.н., профессор, зав. сектором НИИСИ РАН*

Рассматривается актуальная задача функциональной верификации цифровых устройств и инструменты для ее решения – языки спецификаций, библиотеки, стандартизованные методики.

Размер систем на кристалле продолжает расти в геометрической прогрессии, благодаря постоянному развитию технологий. Если в 80-е годы крупные проекты насчитывали десятки тысяч вентиляей, то сегодня увеличение производительности разработки программной и аппаратной составляющих позволяет создавать системы размерами в миллионы вентиляей и строк кода.

Логический синтез, готовые IP-блоки, стандартные интерфейсы, использование старых проектов в новых разработках значительно упростило создание аппаратной части системы, а языки высокого уровня, эффективные оптимизирующие компиляторы и готовые библиотеки функций повысили производительность программистов.

Самым существенным ограничением, с которым столкнулось проектирование цифровых схем на рубеже веков, было связано с функциональной верификацией. В традиционных маршрутах верификации тестирующая программа, созданная для конкретного блока в проекте, могла использоваться в полной мере только для тестирования данного блока. Тестирование блока в сопряжении с другими блоками требовало написания новых тестов. Аналогично, после объединения всех модулей при тестировании всей системы целиком нужно было создавать фактически с нуля и системные тесты. В результате, для верификации большого проекта требовалось в среднем в три раза больше инженеров, чем для его разработки, а верификация, занимая 70% времени и ресурсов в процессе проектирования, не выявляла всех функциональных ошибок, что приводило к необходимости несколько раз выпускать проект.

Большую роль в процессе верификации играет реализация связей между спецификациями проекта, средой разработки и собственно средой тестирования. Необходимо убедиться, что все требования спецификации включены в план тестирова-

ния, а сам проект будет правильно верифицирован. Задача осложняется тем, что требования хранятся в виде множества источников, включая базы данных с требованиями, документы, содержащие установленные требования и таблицы с планами тестирования. В результате требуется много усилий, чтобы выяснить, на что повлияет изменение тех или иных ограничений, или какие требования спецификации нужно верифицировать заново при изменении того или иного блока. Процесс проектирования и верификации традиционно осуществлялся последовательно. Взаимодействия между проектировщиками и верификаторами не были объединены средствами автоматизации в единый поток. Все это еще больше увеличивало сроки и стоимость проектирования.

На преодоление все возрастающих проблем было предпринято широкомасштабное наступление, что привело к радикальному изменению методологии проектирования и времени, затрачиваемому на функциональную верификацию: оно сократилось более чем в четыре раза.

В основу такого качественного прорыва были заложены уже вполне вызревшие идеи и методы. Наибольшее влияние оказало следующее.

Использование парадигмы объектно-ориентированного проектирования, где экземпляры класса обмениваются между собой дискретными порциями информации (транзакциями), позволило поднять уровень описания как самого проекта, так и его тестового окружения на более абстрактный уровень. Экземпляр класса изначально тесно ассоциировался с физической или абстрактной сущностью, аналогичной понятию модуль в иерархической структуре проекта, что еще больше сблизило процессы создания аппаратных и программных средств. Если при реализации проекта необходимо пройти последовательно от абстрактного описания через описание на уровне

регистровых передач, логического представления - к физической реализации, то уровень тестовой среды можно и нужно оставить на прежнем уровне, меняя только уровень интерфейса между средой и проектом.

Другим фактором следует считать парадигму обеспечения управляемости и наблюдаемости объекта проектирования. Идея естественным образом была заимствована из тестопригодного проектирования и дополнена опытом тестирования программного обеспечения (языки спецификаций, метрики покрытий, методики проведения регрессионного тестирования и пр.) и вылилась во внедрение в спецификации, предназначенные для синтеза аппаратуры, спецификаций для проверки свойств этой аппаратуры. В описание стали внедряться как отдельные директивы языков спецификаций типа: *assert* (установи истинность свойства), *restrict* (ограничь входной поток данных), *cover* (собери статистику по выполнимости свойства), так и целые программные блоки (чекеры и мониторы), наблюдающие за выполнением свойств (протоколов функционирования, временных последовательностей, фильтрацией входных данных).

Желание использовать более мощные логики при описании дискретных устройств возникло давно (Pnueli 1977год - Linear-time Temporal Logic, Clarke и Emerson 1981- Computation Tree Logic), но реализовалось оно в реальном процессе проектирования только в начале нашего века, базируясь на языке Property Specification Language, а затем и языке SystemVerilog Assertions. Это позволило создать принципиально новую методологию функциональной верификации (Assertions Base Verification) и использовать ее как при моделировании – динамическая верификация, так и в статике – статическая верификация (формальная верификация).

В настоящий момент базой для проведения верификации проектов являются следующие составляющие.

- Языки описания и спецификации аппаратуры: SystemC, SystemVerilog, Property Specification Language (PSL), SystemVerilog Assertions (SVA)
- Методологии и библиотеки для поддержки практической реализации:
 - *Системный уровень: Transaction-Level Modelling (TLM), Open Verification Methodology (OVM), Advanced Verification Methodology (AVM)*
 - *Уровень RTL: Accellera's Open Verification Library (OVL), Incisive Assertion Library (IAL), Questa Verification Library (QVL), Questa Multiview Verification Component*

(MVC), CheckerWare, Assertions Base Verification (ABV), Assertion Synthesis, Intelligent Testbench Automation, VMM methodology, VCS Verification Library

Организирующим началом проведения функциональной верификации является тестовое окружение (тестбенч), позволяющее эффективно использовать выше перечисленные методики и библиотеки.

Тестовое окружение

К тестовому окружению проекта предъявляются требования:

Скорость – быстрое создание тестбенча для отдельного модуля системы.

Повторное использование – тестбенч модуля должен использоваться во всех проектах без внесения изменений.

Наращивание объекта – созданные отдельные тестбенчи для модулей должны использоваться в качестве частей тестбенчей для нескольких модулей и системы в целом без существенных модификаций.

Выполнить эти требования позволяет создание набора отдельных блоков, автоматизирующих свои задачи, которые подключаются к проекту, формируя целостное окружение проекта (рис. 1).

Сокращения времени разработки окружения проекта можно достичь воспользовавшись реализованными методологиями верификации, которые включают набор базовых библиотечных элементов, реализованных в виде классов на SystemC и SystemVerilog, и набор рекомендаций по их применению. Примером такой методологии является Open Verification Methodology, реализованная на SystemVerilog. Набор классов в OVM покрывает все окружения проекта, показанного на рис.1, и позволяет быстро создавать блоки, пригодные для использования в различных проектах.

Алгоритмический синтез тестов

Алгоритмический синтез тестовых воздействий по заданным правилам - это относительно новый подход, призванный решить проблемы, связанные с такими подходами, как направленные (детерминированные) тесты и случайные тесты с ограничениями. Ранее такая техника использовалась при тестировании программ и в мире аппаратуры попала недавно. Здесь «тестбенч» является на самом деле набором правил. Способ представления этих правил можно назвать грамматикой, а каждое правило – предложением, состоящим из нескольких действий, где каждое действие соот-

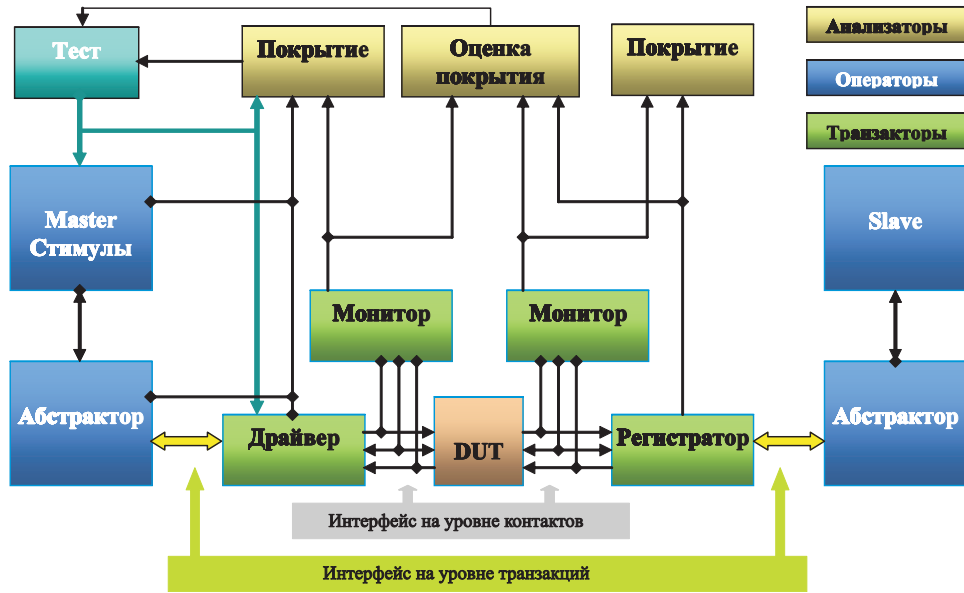


Рис.1. Окружение проекта

ветствует конкретной задаче тестбенча. Программа алгоритмического синтеза тестов читает этот набор правил и под руководством пользователя автоматически генерирует соответствующий тестбенч. Небольшой набор правил синтезируется в большое количество тестов, а пользователь направляет программу, указывая порядок тестирования, различные значения параметров тестов и т. д.

Для пояснения создадим алгоритмический тестбенч для счетчика, в котором три правила Count_Test, Load_Test и Clear_Test будут перебираться во всех возможных комбинациях:

```
Start_Test = my_initial, repeat
```

```
((Count_Test | Load_Test | Clear_Test)),  
final_action;  
Count_Test = count_on, repeat((idle, min  
(1, max(2))), count_off;  
Load_Test = load_random_digit;  
Clear_Test = flush_to_zero;
```

Развернутый вид этого тестбенча показан на рис.2, диапазон [1:8] указывает на количество проходов по ребру.

Отметим, что набор правил для тестирования модуля несоизмеримо мал по сравнению с размерами направленного тестбенча.

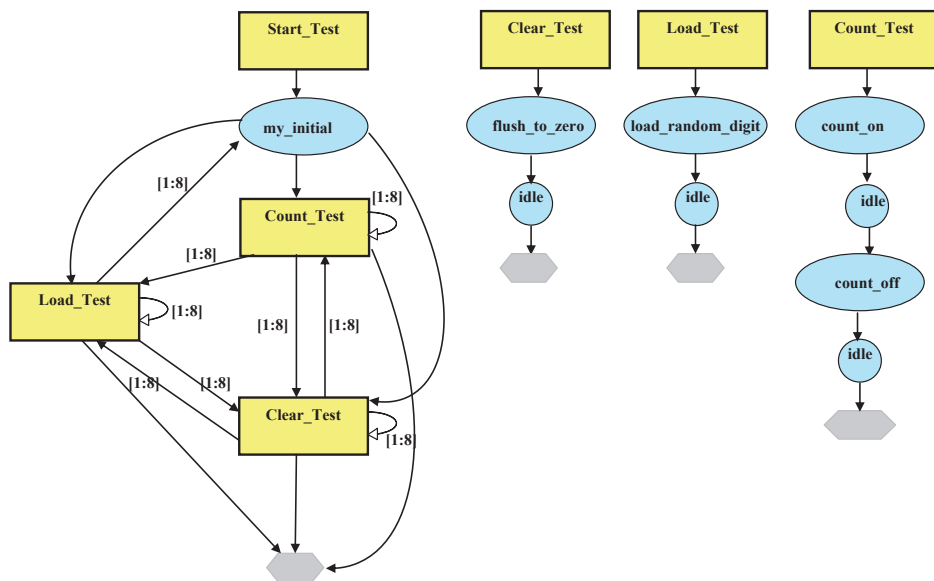


Рис.2. Алгоритмический тестбенч для счетчика

Время для написания и отладки набора правил сравнимо с затратами на случайные тесты с ограничениями, но при этом время моделирования для достижения аналогичного тестового покрытия намного меньше. Правила в тестбенче имеют иерархическую структуру: правила верхнего уровня ссылаются на правила более низких уровней и т. д. Кроме того, правила можно обходить, как видно из нашего примера, по нескольким маршрутам.

Отдельного внимания заслуживает тот факт, что помимо тестирования описания аппаратуры, данная технология может использоваться для автоматического тестирования связи программных и аппаратных частей проекта.

Набор правил для модуля может быть использован для тестирования нескольких модулей или системы в целом, для этого нужно создать «исполняющий» набор правил для управления существующими наборами правил. Это дает возможность создавать верификационные IP, например, для протоколов шин АНВ, AXI, OCP, PCI и спецификаций интерфейсов Ethernet, USB, SONET, UART.

За счет оптимизации прохода по графу алгоритмического тестбенча удается значительно повысить эффективность. Так, если для достижения 70% покрытия, используя случайные тесты с ограничениями, требуется 2 недели, то по методике интеллектуального создания тестбенчей можно достигнуть 100% покрытия на том же проекте всего за 3 часа тестирования.

Assertions Base Verification

Динамическая функциональная верификация связана с решением двух базовых задач. Это - генерация входных последовательностей и фиксация результатов. Последняя задача имеет два аспекта: необходимо зафиксировать корректность функционирования и оценить объем и глубину проверенного функционирования. Решение первой задачи в традиционном тестировании решается на основе сравнения полученных результатов моделирования с ожидаемыми результатами, а вторая связывается непосредственно с кодом описания и/или с объемом данных, пропущенных через модель. Принципиальным моментом в АВВ является тот факт, что описываются и проверяются свойства проекта, которым он должен соответствовать в тех или иных ситуациях, а полнота тестирования оценивается по покрытию этих свойств в процессе моделирования.

Лингвистической основой функциональной верификации служат языки Property Specification Language и SystemVerilog Assertions, имеющие общую семантику. SVA является частью System-

Verilog и предназначен для проведения динамической и статической верификации. В отличие от SVA PSL можно использовать в проектах, представленных на языках VHDL, Verilog, SystemVerilog, GDL, SystemC. Основу PSL и SVA составляют свойства - конструкции, использующие логические, временные операторы, последовательности и множество встроенных функций, и описывающие некоторое поведение будущей аппаратуры («за сигналом запроса через три такта ожидается сигнал ответа» или «счетчик А никогда не переполняется» и т.д.). Далее к сформулированным свойствам применяются различные директивы, что именно необходимо выполнить с этими свойствами в процессе верификации: проверить выполнимость свойства, предположить выполнимость свойства, зафиксировать данное поведение в процессе моделирования и пр.

Для иллюстрации, приведем примеры на PSL:

Пример 1.

Два свойства, присущие D-триггеру (D - данных, RS - сброс в '0', C - синхронизация, Q - выход):

```
property P1 = always not RS -> not Q @ True; -  
всегда при нулевом значении RS следует нуле-  
вое значение Q и property P2 = always rose(C)  
and RS -> Q = D; - всегда при положительном  
фронте C и единичном значении RS Q равен D.  
Директивы проверки: assert P1; assert P2;
```

Пример 2.

Пусть необходимо проверить, что при последовательном срабатывании сигналов req и ack (**sequence** S1 = {req; ack};) должно последовать срабатывание сигналов start, busy и end, причем busy должно удерживаться 3 такта (**sequence** S2 = {start; busy[3]; end};). Свойство, описывающее данное поведение, имеет вид: **property** P3 = **always** S1 ||=> S2; а директива проверки свойства: **assert** P3;

Реализация АВВ приводит к необходимости помимо создания HDL - модели описывать и свойства этой модели. Принципиальным отличием этого описания является то, что свойства описывают не только то, как должны функционировать и взаимодействовать различные устройства, но и как они не должны функционировать. Важным отличием является и различие целей и соответствующие этим целям формы представления. Причем, если проект проходит различные стадии представления, то его свойства фактически остаются неизменными, и только уточняются и детализируются их представления.

Интенсивное использование стандартизованных методик, библиотек и комплексных средств проектирования позволяет повысить качество функциональной верификации сложных устройств и в значительной мере снизить трудоемкость и производственные затраты.

Анализ оптического контроля соединений многослойных печатных плат

Козлов А. С., *вед. электроник*

Институт Системного Анализа РАН

Предлагаются статистические методы анализа параметров МПП, используя метод Монте-Карло. Рассматриваются проблемы контроля и надежности при проверке модулей на основе автоматизированной системы оптического считывания топологии. Дается математическая модель модулей и объектов контроля по параметрам импеданса Z , методика диагностики на основе контрольных точек при проектировании электронных модулей с помощью САПРов Mentor Graphics и Boundary Scan.

Разрабатывая ЭВМ БЭСМ-6, АС-6 академик В.А. Мельников всегда использовал комплексный подход в работе. В результате при проектировании и наладке ЭВМ «СС БИС» был создан уникальный коллектив в НИИ «Дельта» из четырехсот специалистов, состоящий из электронщиков, технологов, химиков, конструкторов, топологов, программистов, математиков и операторов вычислительной техники. Большое значение придавалось контролю и проверке электронных блоков и устройств.

Успешно применялся контроль межсоединений многослойных печатных плат (МПП) при помощи метода четырехзондового контроля [1]. Суть этого метода заключается в последовательном подключении зондов устройства контроля монтажа (УКМ) в контрольных точках по программе управляющей ЭВМ с автоматическим измерением и сопоставлением импедансам Z .

При выборе конструктивно-элементной базы разработчику электронных блоков следует иметь в виду вопросы надежности, контроля, ремонтно-пригодности. Причем, очевидна неэффективность контроля после процесса изготовления печатных плат.[2] В настоящее время контроль закладывается на этапе проектирования при помощи методов САПР: Mentor Graphics (MGC), Boundary Scan (BSA), P-CAD [3].

Известно [4], что при тактовой частоте 10ГГц и выше особо жесткие ограничения накладываются на волновые сопротивления конструкции, поэтому возникает задача определения импедансов Z , связанных с параметрами линии соотношением:

$$Z = \sqrt{(R + j\omega L) / (G + j\omega C)}$$

Недостатком данного метода является принципиальная невозможность выполнить зонды для шага печатных проводников менее 100 мкм. Поэтому был разработан и испытан на реальных МПП оптический метод контроля. [5]. При помощи данного метода по геометрическим размерам проводников и переходов с последующим расчетом электрических параметров определяется Z . Далее по программе сравниваются полученные данные с теоретически рассчитанными в САПРе.

Оптический метод контроля позволяет сравнить форму и геометрические размеры проводников на слоях печатных плат с проектными параметрами, а также рассчитать и сравнить импедансы проводников. При этом может оказаться, что импедансы мало отличаются от расчетных даже при заметных отклонениях геометрических размеров проводников от проектных данных. Различные дефекты изготовления слоев печатных плат могут почти полностью скомпенсировать друг друга в отношении импеданса проводников на плате, имеющих дефекты. Более того, дефекты, занимающие незначительную часть длины проводников, вообще мало отражаются на их импедансе. Такие дефекты обнаруживаются оптическими методами путем прямого сравнения изготовленного слоя с эталоном, а измерения электрических параметров проводников, например, четырехзондовым методом, не определяют дефектов. В то же время, такие случаи создают угрозу замыканий или обрывов проводников печатных плат в процессе эксплуатации. Замыкания возникают в местах утолщения проводников, а обрывы в наиболее тонких местах.

Таким образом, оптические методы контроля не только применимы в случаях, где невозможны прямые электрические измерения, но позволяют прогнозировать безопасность печатных плат в части возникновения дефектов, неопределяемых сразу после изготовления.

Оптические методы послойного контроля МПП являются единственно пригодными для обеспечения надежности и безопасности эксплуатации современных электронных устройств.

Автоматизированная система [6] построена на оптическом считывании изображения с помощью лазера и позволяет получить в конечном итоге волновые сопротивления в контрольных точках.

В результате анализа получается математическая модель:

$$M = \{E, R, D\}, \text{ где}$$

E - множество объектов СПП: переходные отверстия, места посадки интегральных схем, конденсаторов и др. элементов, утолщения на месте разветвления проводников. В свою очередь, объекты имеют характеристики (множество D): величины радиусов, размеры, координаты проводников и отверстий.

Объекты и характеристики связаны определенными отношениями (множество R): соединения отверстий, утолщения, а также пересечения проводников.

Дефектами в данной модели M считаются любые отклонения в токопроводящих рисунках от описанных в разработанной схеме характеристик элементов и отношений между ними: дополнительные отверстия, обрыв проводников, короткие замыкания, пятна (дополнительные токопроводящие рисунки), утолщения или утончения.

Анализ данной математической модели спецпроцессором позволяет сделать вывод о характеристиках слоя по волновым сопротивлениям Z : удовлетворяют ли они требованиям распространения сигналов. Не измеряя электрических параметров слоя непосредственно, удастся получить их при помощи оптического метода. Полученные данные сравниваются с теоретическими в контрольных точках при проектировании методами САПР [7].

Ввиду высокой плотности монтажа МПП постоянно повышается вероятность отказов в виде коротких замыканий («закороток») и разрывов печатных проводников («обрывов»). В процессе прессования МПП под давлением может произойти смещение проводников, что приведет к их обрывам или закороткам. Поэтому необходим контроль МПП в сборе.

Предлагаемый метод теоретического тестирования BSA [8] рассматривает собранную МПП,

исключая комбинационные схемы, которые анализируются на этапе проектирования с помощью MGC или P-CAD отдельно. Используя методы вычислительной геометрии, находятся дефекты смежных соседствующих проводников. Сложность расчетов определяется числом цепей N и числом дефектов L . Для анализа компьютером МПП задаются входные тестовые векторы (ТВ), а в результате определяются критические точки.

Вначале предполагается, что каждая цепь может быть закорочена с любой другой. При тестировании ТВ задаются как входные на всю схему. В этом методе используется постулат соседства, в котором принимается, что в случае соседства линий N_i и N_j они могут закоротиться, и если какие-то N_i и N_j закорочены, то все находящиеся между ними тоже закорочены, а другие линии не учитываются на «закоротку».

Правда, этот постулат не идеален, т.к. практически возможен случай, изображенный на рис.1., где замкнулись несоседние N_i и N_k , а соседние N_i и N_j наоборот не замыкаются. Но это исключение, которое все равно обнаружится позднее по выходному сигналу «Вых.Т».

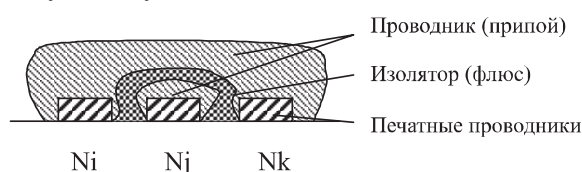


Рис.1. Внешний вид дефекта пайки

На втором этапе производится минимизация ТВ. Анализируются только линии, не охваченные соседством в предыдущих точках. Векторы могут быть, согласно постулата соседства, в пределах списка. Число ТВ для набора списка варьируется в зависимости от топологии схемы. Начальный набор векторов соответствует исходному тесту. Линии, которые не изменяют своего соседства, считаются проверенными и при загрузке новых векторов пропускаются. Затем осуществляется процедура минимизации для уменьшения количества тестовых векторов.

Недостатком метода является трудоемкость выявления точек пересечения V_i . Но следует учесть, что чаще всего схемы соответствуют «манхэттенской» архитектуре, когда состоят из набора горизонтальных и вертикальных отрезков, которые соединены между собой. Для решения такой задачи алгоритм модифицируется при помощи методов вычислительной геометрии и компьютерных программ.

Просмотровый анализ начинается с определения конечных точек, которые соответствуют номеру горизонтальных и вертикальных линий рисунка. Точка пересечения V_i – это точка рисунка, где два отрезка пересекаются, что соответствует изменению порядка соседства двух пересекающихся линий.

При манхэттенской архитектуре отрезки идут один за другим в цепочку с пересечением. Применяется упорядочение и просмотр отрезков по вертикали от одной стороны до другой, попутно производя запись и генерацию новых наборов тестовых векторов. Этот метод базируется на упорядочении цепочки отрезков, тотальном сканировании схемы и определении пересечений с соседними отрезками.

На рис.2а приведен простейший пример схемы, наглядно показывающий методику контроля при помощи BSA. Здесь манхэттенская структура линий N_i , где $i=0,1,2$. Набор ТВ для отрезков обновляется, если меняется соседство линий.

В точках V_i линия просмотра достигает начала или конца отрезков, где список соседей может измениться. Так в линии просмотра «1» точка V_3 – это конец отрезка N_3 , но соседство линий не меняется, поэтому ТВ не изменяется с началом следующего отрезка N_3 .

В то же время V_2 для линии N_3 – точка пересечения с линией N_2 с изменением соседств и соответственно ТВ.

Для упрощения можно изъять точки, не изменяющие ТВ и не влияющие на ход тестирования. Так в данном случае, точки V_1 и V_3 исключаются для линии просмотра «1». Упрощенная схема в этом случае примет вид, показанный на рис.2б.

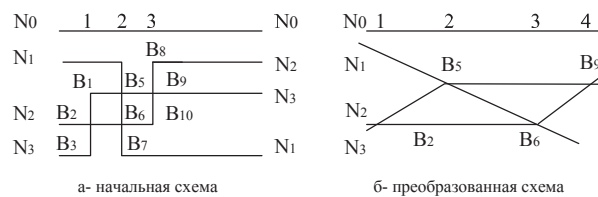


Рис.4. Простейшая схема контроля векторами ТВ.

Вначале происходит поиск начала цепочки и генерация первоначального тестового вектора. В данном примере начало цепочки в списке N_0 - N_1 - N_2 - N_3 , что соответствует нумерации линий схемы.

Таким образом, ТВ для V_2 приобретает вид «У X X X».

Следующим этапом является сканирование и перемещение по схеме от входа к выходу. Аналогично рассматриваются V_5 , V_6 для линии просмотра «2» и V_9 для линии просмотра «3».

Как только обработаны все точки пересечения, генерация прекращается, но некоторые точки

имеют значения неопределенности «У». Поэтому следующим этапом является минимизация, которая служит для сокращения вариантов путем слияния одинаковых и близких векторов. Для табл.1 процесс минимизации сольет векторы V_2 и V_6 . В результате получается табл.2.

Для того чтобы автоматизировать процесс поиска отрезков, событий пересечения и других случаев, приведенных в примере, необходимо формализовать последовательность, т.е. написать алгоритм поиска. Алгоритм – программа поиска приведена в [9].

После определения критических точек методом BSA производятся физические измерения для определения наличия закороток и обрывов в МПП и соответствия заданной схеме.

Анализ можно провести методами теории вероятности, учитывая, что импеданс Z – это волновое сопротивление.

$K(i)$ – крутизна характеристики изменения волнового сопротивления от изменения значения соответствующего параметра.

$DW=W_0 - W$ – отклонение ширины сигнального проводника от его математического ожидания.

$C_w=C(W_0)/C_v(V)$ – коэффициент стоимости, определяющий увеличение затрат на уменьшение поля допуска относительно исходных затрат на существующий допуск.

$Z(W)=K(i)DW C_w$ – вклад частного параметра W в отклонение от номинала выходного параметра.

Коэффициенты чувствительности $K(i)$ представлены как отклонение DZ , например, на 10 мкм от каждого линейного параметра. Эти данные необходимы и их достаточно для написания итерационной части программы вычисления функции качества.

Решение задачи допускового анализа можно получить путем подстановки в функцию изменения качества вместо v_i соответствующих значений допусков (e_i), т.е. в виде $dZ_{max} = \hat{a}(q_i e_i)$.

Решение задачи допускового синтеза [2] при условии, что разброс параметров одинаков и равен e , найдем как $e = dZ_{max} / \sum(q_i)$.

Если же значения разброса параметров различны, то данная задача превращается в задачу оптимизации.

Решение задачи допускового синтеза можно сформулировать как поиск таких значений допусков на параметры, которые обеспечат заданное значение DZ_{max} .

При больших отклонениях параметров от своих номинальных значений пренебречь высшими членами ряда Тейлора невозможно ввиду нелинейного характера функции качества, поэтому целесообразно использовать моделирование технологического разброса методом Монте-Карло [10].

Сущность метода равносильна многократному натурному макетированию конкретной характеристики путем генерирования случайных реализаций. При этом предполагается, что параметры v_i распределены по нормальному закону (распределение Гаусса).

Если в пределах каждого частного поля допуска разброс параметров имеет равновероятное распределение случайной величины, то достаточно будет просуммировать частные вклады каждого параметра в результирующее отклонение выходного параметра, если их число невелико.

В действительности все рассматриваемые параметры подчиняются нормальному закону распределения случайной величины в каждом частном поле допуска. Следовательно, линейного суммирования погрешности не произойдет. Вообще надо знать вероятностные законы функционирования отдельных частей системы параметров. Тогда метод Монте - Карло (метод перебора) позволит определить вероятностные закономерности работы всей системы, как бы сложна она ни была. Это свойство метода чрезвычайно важно в случае анализа качества системы, содержащей элементы с законами распределения плотности вероятности, отличными от равновероятного или нормального. Помимо предложенного использования формул, должна решаться и другая задача: определение функций качества каждого выходного параметра. Так, формула расчета перекрестной наводки включает формулу расчета волнового сопротивления и коэффициент емкостной связи, зависящий от краевых емкостей, т.е. является существенно более сложной. Аналитические расчеты коэффициентов чувствительности (частные производные по всем параметрам) нецелесообразны из-за чрезмерной сложности исходных формул, поэтому важно использовать некоторый универсальный способ - численное дифференцирование в процессе итерационных расчетов на ЭВМ. Следует также учесть, что при большой кривизне исследуемой функции в интересующей нас области и большом поле допуска появляется ошибка экстраполяции. Для устранения отмеченного недостатка целесообразно использовать метод Монте-Карло в каждой из n частей (больше, например, 10) исходного поля допуска при условии, что в каждой из n подобластей кривизна функции существенно низкая.

Следует выделить особенность метода - погрешность вычислений, как правило, пропорциональна $\sqrt{D/N}$, где D - некоторая постоянная, зависящая от выбранного метода, N - число испытаний. Отсюда видно, что для того, чтобы уменьшить погрешность в 10 раз, нужно увеличить N (т.е. объем испытаний) в 100 раз.

Задачу допускового синтеза удастся решить лишь итеративно, т.е. задаются максимальными значениями разброса каждого i -го параметра и выполняются серия испытаний по методу Монте-Карло.

Рассмотренный метод представляет собой метод адаптивной стохастической оптимизации, используемый в самонастраивающихся математических методах поиска оптимальных параметров систем различной физической природы.

Для моделирования конкретной величины разброса каждого i -го параметра необходимо предусмотреть метод реализации равновероятного распределения случайной величины, а затем на его основе - распределение случайной величины по нормальному закону.

Равномерное распределение случайной величины h , сосредоточенной в интервале от a до b , определяется по известной формуле

$$(1) \quad v = \int_a^b P(x) dx,$$

где $P(x)$ - функция распределения плотности вероятности.

Если функция распределения плотности вероятности $P(x)=1/(b-a)$, то функция η будет представлять равномерное распределение случайной величины.

Действительно, $P(x)=1/(b-a)$ - вероятность случайной величины x в интервале $(b-a)$ и $a < x < b$

$$\int_a^{\eta} \frac{dx}{b-a} = v, \quad \frac{\eta-a}{b-a} = v, \quad \eta = a + v(b-a).$$

В данном случае получена зависимость η , характеризующая отклонение от математического ожидания a в допуске $(b-a)$ с равномерным стохастическим изменением разброса, подчиняющегося случайной величине v .

Большинство анализируемых геометрических параметров высокочастотных элементов подчиняется нормальному закону распределения случайной величины и поэтому рассмотренный равномерный закон распределения необходимо связать с функцией Гаусса, поскольку формула вида (1) является базовой, а входящая в нее функция $P(x)$ в общем случае может быть произвольной.

Для моделирования случайной величины в пределах ожидаемого отклонения от номинала необходимо ввести в формулу (18) соответствующий закон распределения.

$$\text{Тогда} \quad \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\xi} e^{-\frac{t^2}{2}} dt = v$$

Приведенная формула представляет собой хорошо известный интеграл вероятности при ну-

левом значении a математического ожидания и значении среднеквадратичного отклонения σ , равного единице.

Моделирование случайных величин с нормальным законом распределения обусловлено спецификой способа моделирования, в котором, в отличие от условий, где распределение Гаусса (или другой закон распределения) задано, для каждого рассматриваемого параметра мы должны вводить в модель конкретный закон распределения случайных величин.

В дальнейшем на каждом шаге (итерации) необходимо менять ширину поля допуска каждого параметра, обеспечивать некоррелированную стохастичность процесса в границах каждого допуска и, используя свойства сходимости, устремить результирующее значение частных разбросов к оптимальному значению допуска выходного параметра.

С учетом предложенного метода и с введением в систему полярных координат (подробности преобразований опущены), окончательные формулы имеют вид:

$$f = (2\ln(1/z_1))^{0.5} \cos(2pz_2);$$

$$h = (2\ln(1/z_1))^{0.5} \sin(2pz_2).$$

Эти формулы позволяют по двум равномерно распределенным числам z_1 и z_2 вычислить (смоделировать) два независимых значения f и h произвольной нормально-распределенной случайной величины с параметрами $a = 0$ и $s = 1$.

Из изложенного выше следует, что расчет допусков технологических параметров проектируемых изделий по предложенной методике вполне осуществим. Для этого необходимо знать вероятностные характеристики всех элементов технологического процесса и уметь вычислять интересующую нас величину как функцию от параметров этих элементов. Тогда случайность параметров можно учесть путем моделирования по соответствующей программе.

Выводы

Для контроля соединений в печатных платах и чипах целесообразно применять методы контроля непосредственно при их изготовлении при помощи оптической системы с последующим анализом спецпроцессором, т.к. механические приборы с этой задачей не справятся.

Результаты оптического тестирования слоев МПП путем математической обработки приводят к фактическим значениям электрических характеристик цепей в контрольных точках и сравниваются с заранее рассчитанными значениями этих характеристик при проектировании МПП.

Метод тестирования BSA позволяет создать таблицу для поиска анализируемых отрезков, соседств, пересечений проводников. Тестовые переменные, полученные в результате применения BSA, контролируются спецпроцессором.

Разработан новый метод расчета, в котором учтены диэлектрическая проницаемость, погонные и краевые емкости, а также отклонения, связанные с допусками на размеры проводников.

На примере расчета высокочастотных параметров МПП показывается целесообразность моделирования разброса физических параметров случайных процессов методом Монте-Карло.

Зависимость отклонения от математического ожидания параметров в допуске с равномерным стохастическим изменением подчиняется закону Гаусса с $s = 1$.

Литература

1. «Автоматизация визуального контроля печатных плат» («Радиоэлектроника за рубежом» 2005г., вып.2, стр.16-18).
2. Стешенко В.Б. «P-CAD – технология проектирования печатных плат» («БХВ-Петербург», 2003г., стр. 6-12)
3. Разевиг В.Д. «Средства проектирования радиоэлектронных устройств» (М, «Телеком», 2002г., стр. 5 – 23)
4. Козлов А.С., Щербаков С.А. Исследование метода оценки надежности в задачах проектирования электронного оборудования. Вопросы теории безопасности и устойчивости систем. Вып.9, М: ВЦ РАН, 2007, с.214-216.
5. Ашуров Г.Д. «Автоматизированное устройство контроля качества рисунка печатных плат» (Сб. науч. тр. МИЭТ, 2002г., стр. 151-158).
6. Козлов А.С., Щербаков С.А. Оптические методы контроля соединений на печатных платах и безопасность эксплуатации электронных устройств. Вопросы теории безопасности и устойчивости систем. Вып.8. М: ВЦ РАН, 2006, с.188-192.

Краткий очерк развития САПР ТОПТРАН (от 104-й серии ИС до суперЭВМ «Электроника ССБИС»)

Водолазский В. И., *к.т.н., с.н.с.*

ВЦ им. А.А.Дородницына РАН

Автор был участником и свидетелем зарождения советской микроэлектроники, принимал активное участие в проектировании первых отечественных интегральных схем (ИС), начиная со 104-й серии (1968 г.). Участвовал в работах по применению первых отечественных микроЭВМ серии «Электроника» для проектирования ИС. Имеет опыт по эксплуатации и ремонту различного технологического оборудования, которое использовалось для проектирования и производства фотошаблонов ИС и многослойных печатных плат (МПП). Более 20 лет посвятил разработке и применению САПР для автоматизированного проектирования топологии цифровых и аналоговых интегральных и гибридных схем, керамических и многослойных печатных плат, включая для суперЭВМ «Электроника ССБИС» (1986 г.). В пределах объема статьи излагается краткая история эволюции отечественной автоматизированной подсистемы топологического проектирования ИС и МПП.

Ключевые слова: микроэлектроника, топология, фотошаблон, маска, САПР, кремниевая компиляция.

При создании автоматизированных систем топологического проектирования [1, 2], как правило, реализуется следующий принцип. В памяти ЭВМ хранятся массивы исходных данных типовых топологических решений, которые по специальным директивам размещаются по заданным координатам, а при необходимости над ними осуществляются различные геометрические преобразования [3-6]. На экране монитора можно отображать топологию либо всей схемы, либо отдельных компонентов и фрагментов с необходимой степенью детализации. Предоставляется возможность добавлять, стирать, изменять и перемещать отдельные элементы и компоненты топологии. При этом изменения вносятся во все слои, содержащие редактируемые элементы [7]. Большое внимание уделяется программам синтаксического контроля исходных данных и программам контроля технологических зазоров между элементами [1].

Исходной информацией для координатографов являются координаты вершин контуров топологии фотошаблона, а исходной информацией для МФНУ - координаты прямоугольников, составляющих топологию фотошаблона. Простейший транслятор осуществляет перекодировку координат в команды управления технологическими устройствами и требует для своей программной реализации нескольких сот команд микро ЭВМ. Минимальный объем оперативного запоминающего устройства (ОЗУ) микро ЭВМ серии «Электроника» составлял не менее 8 тыс. слов. Этот объем ОЗУ позволил хранить в памяти ЭВМ 6-7 тыс. координат и был достаточен для

представления и трансляции данных топологии фотошаблона ИС средней степени интеграции [8].

Представление топологии фотошаблона координатами всех контуров или прямоугольников является избыточным, так как топологию интегральной схемы можно представить номенклатурой незначительного числа компонентов, а БИС используют регулярную топологическую матричную структуру [9], элементы которой повторяются на большей площади кристалла. Для исключения избыточности исходной информации используют языки описания топологии [5,10], которые однозначно определяют топологию меньшим количеством исходных данных.

Сокращение количества исходных данных еще не означает сокращения объема ОЗУ, необходимого для трансляции управляющей информации с языка описания топологии. Во-первых, для хранения исходных данных библиотеки типовых топологических решений даже небольшого числа компонентов требуется несколько тысяч слов, так как каждый компонент содержит исходные данные не одного слоя, а всей совокупности слоев ИС. Во-вторых, программное обеспечение должно не просто транслировать исходные данные библиотечных компонентов, но и осуществлять различные геометрические преобразования топологии компонентов, которые заданы по определенным правилам языка описания. В третьих, транслятор, как правило, включает необходимые программы для семантического контроля и редактирования макро описания топологии. Таким образом,

транслятор с языка описания потребует несколько десятков тысяч слов ОЗУ ЭВМ. Объем ОЗУ в 28 тыс. слов микроЭВМ серии «Электроника» был достаточен для трансляции управляющей информации фотошаблона БИС с интеграцией в несколько тысяч компонентов [2, 8].

Редактирование топологии в автоматизированных системах с использованием транслятора осуществляется на языке описания и требует многократной прорисовки совмещенного чертежа кристалла. Применение языков описания повышает производительность проектирования топологии в несколько раз. Основным преимуществом трансляторов является их простота и невысокая стоимость разработки и эксплуатации, возможность использования их для проектирования ИС всех типов. Главным недостатком трансляторов является необходимость неоднократной прорисовки совмещенных чертежей с последующим визуальным контролем и исправлением ошибок описания топологии. Все это снижает оперативность проектирования топологии.

Следующий этап совершенствования системы проектирования связан с использованием интерактивных систем на базе графических дисплеев [7-11]. В таких системах проектирование осуществляется с использованием языка описания, а графические мониторы применяются для быстрого визуального контроля и редактирования топологии, что исключает многократную прорисовку совмещенных чертежей и в несколько раз повышает оперативность процесса проектирования.

Графические мониторы - эффективное техническое устройство для редактирования топологии, но требуют разработки для этих целей соответствующих средств программного обеспечения. Первая задача связана с разработкой алгоритмов и программ для фрагментации [13] целостного изображения фотошаблона с целью визуализации отдельных частей шаблона в определенном масштабе. Были разработаны программы типа «скользящее окно» и «скользящая лупа», реализующие алгоритм фрагментации. В некоторых автоматизированных системах, использующих программы фрагментации, для редактирования топологии стали устанавливаться по два графических дисплея на одно рабочее место: на одном дисплее отображается с незначительным увеличением изображение всей топологии и контур зоны визуализации, а на другом — увеличенное изображение фрагмента, ограниченного контуром визуализации на первом дисплее. Редактирование топологии по-прежнему оставалось на языке описания, а графические дисплеи использовались, по существу, в качестве быстрого прорисовщика [12].

Программное обеспечение автоматизированных дисплейных систем с фрагментацией изображения

составляет десятки тысяч команд и требует для формирования, хранения и визуализации промежуточной информации фрагмента достаточно развитой операционной системы. В автоматизированных системах с фрагментацией изображения подпрограммы и файлы данных топологии хранятся на дисковом запоминающем устройстве и вызываются каждый раз в ОЗУ ЭВМ при выполнении заданной операции фрагментации, визуализации или редактирования топологии. Время выполнения таких операций составляет 30 - 60 с, что значительно меньше времени выполнения контрольных прорисовок, но достаточно велико для процесса интерактивного контроля и редактирования топологии.

Решение задачи интерактивного редактирования изображения непосредственно на экране дисплея привело к необходимости трансляции с исходного языка некоторого внутреннего представления данных топологии, которые однозначно связаны с каждым редактируемым элементом. В этом случае редактирование изображения на экране приведет к изменениям данных внутреннего представления, никак не отражающихся в исходном описании топологии. При таком процессе интерактивного редактирования роль исходного описания топологии снижается, так как оно используется только для генерации изображения или для трансляции внутреннего представления данных, которые редактируются в интерактивном режиме независимо от исходного описания [14].

Кодирование топологии, ввод, синтаксический контроль и редактирование исходных данных являются достаточно трудоемкими процессами. Поэтому использование языка описания для одноразовой операции генерации изображения становится нецелесообразным. Предпочтение отдается таким техническим средствам формирования изображения, как световое перо, маркер, сенсорная панель. В этом случае, используя дискретную сетку, разработчик создает непосредственно на экране различные графические примитивы, из которых формирует целостное изображение проектируемой топологии. Процессы редактирования изображения на экране дисплея и преобразование исходных координат во внутреннее представление базы данных осуществляется под управлением программ, совокупность которых образует программное обеспечение автоматизированных систем машинной графики. Использование машинной графики требует применения супермини ЭВМ с большим быстродействием и значительным объемом ОЗУ, а также дисковых запоминающих устройств большой емкости. Процесс проектирования в системах машинной графики осуществляется только с использованием технических средств, затраты на эксплуатацию которых достаточно велики

[Система «Кулон»]. Такова эволюция развития интерактивных автоматизированных систем проектирования топологии БИС и печатных плат, сложившаяся к началу 80-х прошлого века.

Каковы же альтернативные пути развития автоматизированных систем на базе использования языка описания топологии?

Расширение возможности языка для описания с помощью одного макро оператора топологии функциональных узлов БИС, заданных таблицами истинности;

Разработка программ семантического контроля макро описания для автоматического поиска ошибок топологии;

Разработка программного обеспечения для трансляции топологии БИС непосредственно с языка макро описания, минуя внутреннее координатное представление данных в памяти ЭВМ;

Разработка специализированного графического редактора для автоматического поиска по графическому изображению адрес-меток макро описания редактируемого элемента топологии.

К середине 80-х в САПР электронного приборостроения произошел резкий поворот в сторону расширения ассортимента специализированных языков проектирования.

Так, Японская телеграфная ассоциация проводит работы по созданию и внедрению Языка Графического Описания (ЯГО) топологии СБИС [10, с.176], используемого в качестве стандартного интерфейса данных языка функционального описания СБИС [10, с.149].

В соответствии с тенденциями создания языков визуализации была принята концепция 3-х уровней языка графического описания, используемого в качестве стандартного графического интерфейса данных языка VHDL (Very Hierarchical Description Language):

- 1-й (высший) уровень — уровень чтения и восприятия человеком. Примером языка этого уровня является язык SMF (Standart Mask pattern Format) Японской телеграфной ассоциации [10, с.176]. Разработка этого языка была тесно связана с языком VHDL.
- 2-й уровень - уровень описания и преобразования структуры графических объектов, обеспечивающий обмен данными между ЭВМ различных типов. Это язык объектных графических файлов, который посредством компоновщика (LINKER - в англоязычной терминологии по информатике) объединяет структурные иерархические объекты в целостную графическую систему. В этом языке мнемонические ссылки на структурные иерархические компоненты представлены абсолютными и относительными адресами.

- 3-й (низший) уровень - уровень побитового отображения графической информации, воспринимаемой конкретными графическими и технологически!» системами. Примером языка этого уровня может быть SOURCE—формат системы «Кулон», форматы управляющих кадров МФНУ ЭМ-549 и т.д.

При этом использовалась следующую тактику разработки и внедрения графического интерфейса языка VHDL:

- в качестве языка низшего уровня был принят SOURCE-формат системы «Кулон», который фактически являлся отраслевым стандартом;
- в качестве языка второго уровня использовался язык символьных объектных файлов, в которых иерархическая структура графических объектов представлена совокупностью абсолютных и относительных адресов взаимных ссылок. Объектные графические файлы могут допускать включения текстовых пояснений только в качестве комментария. Результатом компоновки и трансляции объектных файлов должна быть целостная топология проектируемой системы в SOURCE-формате. Такой компоновщик объектных файлов может быть поставлен на отечественных микро-ЭВМ типа «Электроника МС1212», ПЭВМ «Электроника-85» и др.
- в качестве языка высшего уровня рекомендовался SMF (Standart Mask pattern Format) – язык графического описания Японской телеграфной ассоциации, который фактически стал стандартным графическим интерфейсом VHDL. Для создания и эксплуатации ассемблера с этого языка потребуются значительные ресурсы мини-суперЭВМ. Результатом трансляции (ассемблирования) описания графических объектов на языке SMF может быть совокупность объектных файлов, которая в последующем компонуется (линкуется) в целостный графический объект проектирования. К созданию такого ассемблера необходимо приступить после успешной разработки и эксплуатации компилятора объектных файлов языка 2-го уровня.

Такова идеология и тактика разработки стандартного графического интерфейса языка VHDL.

Возникает вопрос, нельзя ли перейти от автоматической генерации отдельных компонентов и фрагментов к автоматической генерации всего проекта. Такой переход возможен на основе кремниевых компиляторов (SiComp) [16, с. 131].

Появление термина «Кремниевый компилятор» обязано некоторым сходством процесса проектирования БИС и разработки программного обеспечения, т.е. некоторой общностью между обычными про-

граммными и кремниевыми компиляторами. Проблемы лексического анализа и грамматического разбора идентичны для обоих компиляторов. Языки кремниевых компиляторов широко используют алгоритмы параметрического расширения аналогичные макрорасширениям в программных компиляторах. Применение кремниевых компиляторов позволяет процесс проектирования свести к процессу программирования проекта, абстрагируясь от представления объекта на детальном уровне.

Важной особенностью процесса кремниевой компиляции является возможность модификации и адаптации компилируемого кода проекта к параметрам технологического процесса, которые, как правило, включают поправочные коэффициенты погрешностей. Кремниевая компиляция также предполагает наличие топологического редактора для «тонкой подчистки» отдельных элементов топологии проекта. Выходным продуктом кремниевого компилятора является описание слоев топологии для изготовления фотошаблонов, что предполагает наличие этапа производства, что бы проект стал изделием.

Кремниевый компилятор можно охарактеризовать и как САПР, с помощью которой можно получить топологию БИС, требующую от разработчика сравнительно небольшого объема данных [15]. Конструкторские задачи в такой системе упрощаются благодаря применению структуры элементарных ячеек, которые образуют библиотеку типовых топологических решений.

Эволюция развития САПР нашла свое явное отражение в версиях программного обеспечения системы ТОПТРАН [12], которая в наибольшей степени отражала на каждом этапе развития возможности компьютерного и технологического оборудования для решения поставленных временем проблем электронного приборостроения и микроэлектроники.

Литература

1. Никишин В.И., Водолазский В.И. Основные аспекты построения системы автоматизированного проектирования на базе мини-ЭВМ. // Вопросы кибернетики. Системы автоматизированного проектирования интегральных схем. М.: НСК, 1984. С. 4-14.
2. Водолазский В.И. Система проектирования ТОПТРАН. // Программные продукты и системы, 1996. №4. с.74-81.
3. Никишин В.И., Водолазский В.И. Система команд геометрических преобразований топологии компонентов. // Вопросы кибернетики. Системы автоматизированного проектирования интегральных схем. М.: НСК, 1984. С. 55-67.
4. Михайлов О.А., Беляева Н.В., Ермолаев В.Н. Система машинного описания топологии и изготовление фотошаблонов ИС. // Электронная промышленность, 1975, №6. С.12-17.
5. Фарина Д., Даффи Дж.Р., Калгрэн Т.Л. Метод библиотечных элементов, позволяющих разработчику любой квалификации проектировать интегральные схемы. // Электроника, 1981. № 24. С. 55-59.
6. Никишин В.И., Водолазский В.И. Язык описания топологии системы ТОПТРАН. // Вопросы кибернетики. Системы автоматизированного проектирования интегральных схем. М.: НСК, 1984. С. 15-33
7. Майкл Диккенс, Ларри Длри. Средства графического редактирования, обеспечивающие построение стандартных элементов и символическое представление подсистем. // Электроника, 1981. № 24. С. 48-54.
8. Попов В.А., Водолазский В.И. Автоматизированная система проектирования топологии шаблонов БИС. // Автоматизированные системы управления. Харьков: ХАИ, 1977, вып. 1. С. 19-24
9. Лысенко Э.В., Водолазский В.И. Программирование преобразования топологии матричных структур БИС с функциональной нерегулярностью. // Сб. Радиоэлектроника летательных аппаратов», вып.10. Харьков, ХАИ, 1978, с.115-121.
10. Киносита К., Асада К., Карацу О. Логическое проектирование СБИС: Пер. с япон. М.: Мир, 1988. 309 с., ил.
11. Баранов Б.И., Гуревич Д.З. Использование средств машинной графики в системе автоматизации проектирования фотошаблонов БИС. // УСиМ, 1976. № 1. С. 92-95.
12. Никишин В.И., Водолазский В.И. Система проектирования ТОПТРАН. М.: Наука, 1988, 112с.
13. Лысенко Э.В., Водолазский В.И. Алгоритм масштабного преобразования топологии шаблонов интегральных схем для электронно-лучевых установок. // Радиоэлектроника летательных аппаратов. Тематический сборник научных трудов ХАИ. Выпуск 9. / Харьков, 1977. С.36-40.
14. Казенов Г.Г., Баталов Б.В., Щербаков В.Е., Еремин Л.В. Автоматизированная система проектирования фотошаблонов на базе ЭВМ БЭСМ-6. // Микроэлектроника, 1976, вып. 9. С.22-26.
15. Автоматизация проектирования БИС. В 6 кн.: Практик. Пособие. Кн. 1. Казеннов Г.Г., Соколов А.Г. Принципы и методология построения САПР БИС. / Под ред. Г.Г.Казенова. М.: Высш. Шк., 1990. 142 с.: ил.

Методы и средства защиты программных продуктов САПР

Смирнов В. Ю. *к.т.н., доцент, заведующий кафедрой*

МАИ

Рассматривается проблема создания комплексной системы защиты программных продуктов САПР. Представляется программный комплекс, который решает задачи защиты продуктов САПР от несанкционированного доступа и копирования с использованием различных методов и средств.

Проблемы защиты данных наиболее остро проявляются при использовании ЭВМ для обработки и хранения информации. Любая современная проектная организация не в состоянии успешно развиваться и вести хозяйственную деятельность без создания на ней условий для надежного функционирования системы защиты собственной информации. И поэтому проблема создания комплексной системы защиты программных продуктов САПР, а также изучение, обобщение и систематизация основных направлений и методов защиты информации, которые реально используются или могут быть использованы в ближайшее время в проектных организациях, несомненно, является актуальной.

Обеспечение безопасности информации в организации – это непрерывный процесс, заключающийся в обосновании и реализации наиболее рациональных методов, способов и путей совершенствования и развития системы защиты, а также непрерывном контроле ее состояния, выявлении ее узких и слабых мест и противоправных действий.

Для решения вышеуказанных проблем были поставлены следующие задачи:

- защита файлов с помощью методов криптографии;
- защита от несанкционированного копирования программных продуктов;
- защита от несанкционированного доступа к файлам.

Для решения задачи шифрования файлов целесообразно использовать современные криптографические системы, обладающие повышенной стойкостью: для алгоритмов шифрования и дешифрования крипто-графическую систему DES,

для создания и проверки электронной подписи – RSA.

Для защиты от несанкционированного копирования файлов программных продуктов САПР необходимо использовать идентификационные характеристики компьютер-ных носителей данных, атрибуты файлов, а также создать свой формат данных, обеспечивающий создание файла только для просмотра результатов расчётов.

Защиту от несанкционированного доступа к информации целесообразно осуществлять с помощью:

- использования метода контрольных сумм;
- применения метода “циклического контрольного кода”;
- использования однонаправленных функций хэширования, которые обеспечивают контроль целостности данных.

В соответствии с поставленными задачами была разработана программа, реализованная под операционную систему Microsoft Windows XP Professional на базе Visual C++, с возможностью встраивания в программные продукты САПР типа AutoCAD. На рис. 1 схематически показана структура программы.

В разработанной программе реализованы следующие способы защиты файлов и программных средств САПР:

- ✓ производится шифрование файлов с возможностью выбора метода шифрования: DES – обладает высоким уровнем надёжности и криптостойкости, а XOR и Гаммирование популярны благодаря своей высокой скорости шифрации;
- ✓ выполнена защита от несанкционированного копирования программных продуктов на основе

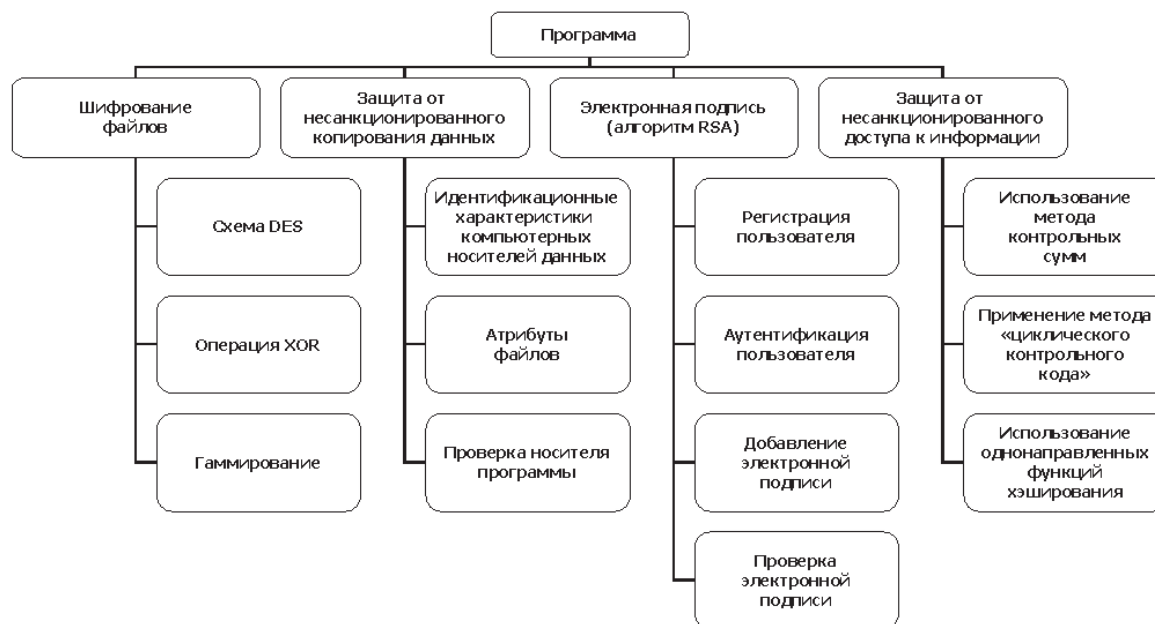


Рис. 1. Структура программы.

- идентификационных характеристик носителя данных и атрибутов файлов;
- ✓ реализована возможность добавления и проверки электронной подписи к файлу;
 - ✓ реализована защита от несанкционированного доступа путем использования метода контроля целостности данных.

В настоящее время, в связи со сложным характером взаимоотношений на рынке программных продуктов САПР, проблема защиты от несанкционированного копирования является одной из наиболее острых в области разработки программных средств. Она обусловлена самой сутью человеческой психологии и будет существовать до тех пор, пока программный продукт является товаром.

Защита от копирования реализована в момент запуска программы. Если данный программный продукт был нелегально скопирован, то работа автоматически завершится. Следует заметить, что в программном комплексе методом защиты от несанкционированного доступа является использование идентификационных характеристик, например, диска. Во время запуска программы по защите данных осуществляется проверка различных свойств носителя данных. В частности, происходит проверка серийного идентификационного номера устройства хранения данных (диска, дискеты, CD или «флэш-памяти»). И в случае несовпадения системных характеристик программа прекращает свою работу. Для надежности работы

такой проверки и от защиты программы от взлома серийный идентификационный номер не просто сравнивается с некоторым номером, забытым в программу, а оба этих номера зашифрованы.

В программном комплексе реализована возможность добавления и проверки электронной подписи к файлу с помощью системы шифрования RSA, а также разработан собственный формат данных для защиты файлов от несанкционированного доступа. При открытии файлов данные преобразуются и отображаются в рабочем окне, а при сохранении на диск шифруются и сохраняются, что обеспечивает недоступность для редактирования данных и переноса их в другие редакторы.

Основным экономическим эффектом от внедрения разработанного программного комплекса, является существенное уменьшение материального ущерба вследствие реализации каких-либо существующих угроз информационной безопасности. Например, за счет создания препятствий для распространения нелегальных копий разработанного программного продукта САПР или предотвращение доступа конкурирующим организациям к перспективным проектам организации.

Для оценки экономической эффективности системы защиты информации организации используются такие показатели эффективности, как отдача на инвестиционный капитал (ROI), совокупная стоимость владения (TCO) и другие. По экспертным оценкам внедрение систем защиты

информации может дать экономический эффект на 40% превышающий вложения.

Далеко не весь ущерб от реализации угроз информационной безопасности можно однозначно выразить в денежном исчислении в текущий момент времени. Например, причинение урона интеллектуальной собственности организации может привести к таким последствиям, как потеря позиций на рынке или потеря постоянных и временных конкурентных преимуществ, что, в конечном счете, также повлечет за собой материальный ущерб, но в будущем. Таким образом, обеспечение информационной безопасности компании имеет вполне конкретный экономический смысл. А достижение этой цели должно осуществляться экономически оправданными мерами.

Выводы

Для обеспечения требуемого уровня информационной безопасности программных продуктов САПР требуется комплексный подход к защите информации. Безопасность информации может быть обеспечена лишь при комплексном использовании всего арсенала имеющихся средств защиты во всех структурных элементах производственной системы и на всех этапах технологического цикла обработки информации. Наибольший эффект достигается тогда, когда все используемые средства, методы и меры объединяются в единый целостный механизм – систему защиты информации. Для реализации комплексной информационной защиты САПР целесообразно использовать современные криптографические системы, обладающие повышенной надёжностью, криптостойкостью и быстродействием.

Разработанная программа, осуществляющая комплексную защиту информации в проектных организациях, позволяет:

выполнять шифрование файлов с возможностью выбора метода шифрования;

осуществлять защиту от несанкционированного копирования программных продуктов с помощью идентификационных характеристик носителя данных и атрибутов файлов;

обеспечивать защиту от несанкционированного доступа к информации с помощью применения методов контрольных сумм, циклического обратного кода и однонаправленных функций хэширования;

добавлять к файлам электронную подпись и проверять наличие электронной подписи у файла; Создан собственный формат данных, обеспечивающий различные виды защиты, в том числе невозможность просмотра файла в других программах, создание файла только для

просмотра без возможности переноса содержимого файла в другие программы.

Программа может быть встроена в AutoCAD, Nastran и другие продукты САПР.

Разработанная программа защищена от копирования и использования нелегальных копий на других носителях данных.

Созданная комплексная информационная защита систем автоматизации проектирования может быть использована для обеспечения компьютерной безопасности САПР во всех проектных организациях.

Внедрение разработанного программного комплекса может дать существенный экономический эффект за счет предотвращения реализации угроз информационной безопасности.

Литература

1. Арзуманов С.В. Оценка эффективности инвестиций в информационную безопасность. – М.: Защита информации. Инсайд, №1, 2005.
2. Губенков А.А. Информационная безопасность. – М.: Новый издательский дом, 2005.
3. Домашев А.В., Попов В.О., Правиков Д.Н. Программирование алгоритмов защиты информации. – М.: НОЛИДЖ, 2000.
4. Зегжда Д.П., Ивашко А.М. Основы безопасности информационных систем. – М.: Телеком, 2000.
5. Зима В.М., Ломако А.Г., Ростовцев Ю.Г. Технологии обеспечения информационно-компьютерной безопасности. – СПб.: ВИКУ, 2000.
6. Конеев И.Р., Беляев А.В. Информационная безопасность предприятия. – СПб.: БХВ-Петербург, 2003.
7. Ярочкин В. И. Информационная безопасность. – М.: Междунар. отношения, 2004.

нормы выбросов по двум видам топлива для автомобилей разной массы;

Дизель		СО	НС	НС+NO x	NO x	ТЧ
(N1, Класс I/1000)*7	Евро 1	2,72	0,00	0,97	0,00	0,14
≤1305 кг	Евро 2, IDI	1,00	0,00	0,70	0,00	0,08
	Евро 2, DI	1,00	0,00	0,90	0,00	0,10
	Евро 3	0,64	0,00	0,56	0,50	0,05
N1, Класс II	Евро 1	5,17	0,00	1,40	0,00	0,19
1305-1760 кг	Евро 2, IDI	1,25	0,00	1,00	0,00	0,12
	Евро 2, DI	1,25	0,00	1,30	0,00	0,14
	Евро 3	0,80	0,00	0,72	0,65	0,07
N1, Класс III	Евро 1	6,90	0,00	1,70	0,00	0,25
>1760 кг	Евро 2, IDI	1,50	0,00	1,20	0,00	0,17
	Евро 2, DI	1,50	0,00	1,60	0,00	0,20
	Евро 3	0,95	0,00	0,86	0,78	0,10
Бензин						
N1, класс 1	Евро 1	2,72	0,00	0,97	0,00	0,00
≤1305 кг	Евро 2	2,20	0,00	0,50	0,00	0,00
	Евро 3	2,30	0,20	0,00	0,15	0,00
	Евро 1	5,17	0,00	1,40	0,00	0,00
1305-1760 кг	Евро 2	4,00	0,00	0,65	0,00	0,00
	Евро 3	4,17	0,25	0,00	0,18	0,00
	Евро 1	6,90	0,00	1,70	0,00	0,00
>1760 кг	Евро 2	5,00	0,00	0,80	0,00	0,00
	Евро 3	5,22	0,29	0,00	0,21	0,00

Нормы выбросов за месяц всеми машинами, зарегистрированными в Москве(30 дней * норма выброса * количество литров сжигаемых в месяц).

		СО	НС	НС+NO x	NO x	ТЧ
(N1, Класс I/1000)*7	Евро 1	31140273,60	0,00	11105171,10	0,00	1602808,20
≤1305 кг	Евро 2, IDI	11448630,00	0,00	8014041,00	0,00	915890,40
	Евро 2, DI	11448630,00	0,00	10303767,00	0,00	1144863,00
	Евро 3	7327123,20	0,00	6411232,80	5724315,00	572431,50
N1, Класс II	Евро 1	78919222,80	0,00	21370776,00	0,00	2900319,60
1305-1760 кг	Евро 2, IDI	19081050,00	0,00	15264840,00	0,00	1831780,80
	Евро 2, DI	19081050,00	0,00	19844292,00	0,00	2137077,60
	Евро 3	12211872,00	0,00	10990684,80	9922146,00	1068538,80
N1, Класс III	Евро 1	26331849,00	0,00	6487557,00	0,00	954052,50
>1760 кг	Евро 2, IDI	5724315,00	0,00	4579452,00	0,00	648755,70

(Продлжение) Нормы выбросов за месяц всеми машинами, зарегистрированными в Москве...

		СО	НС	НС+NO _x	NO _x	ТЧ
	Евро 2, DI	5724315,00	0,00	6105936,00	0,00	763242,00
	Евро 3	3625399,50	0,00	3281940,60	2976643,80	381621,00
		30,00				
N1, класс I	Евро 1	170190590,40	0,00	60692967,90	0,00	0,00
≤1305 kg	Евро 2	137654154,00	0,00	31285035,00	0,00	0,00
	Евро 3	143911161,00	12514014,00	0,00	9385510,50	0,00
N1, Класс II	Евро 1	395373320,10	0,00	107064342,00	0,00	0,00
1305-1760 кг	Евро 2	305898120,00	0,00	49708444,50	0,00	0,00
	Евро 3	318898790,10	19118632,50	0,00	13765415,40	0,00
N1, Класс III	Евро 1	719555805,00	0,00	177281865,00	0,00	0,00
>1760 кг	Евро 2	521417250,00	0,00	83426760,00	0,00	0,00
	Евро 3	544359609,00	30242200,50	0,00	21899524,50	0,00

Используя симплекс-метод для решения, проведем вычисления с помощью системы компьютерной математики **Derive 6.1**, получаем конкретные результаты нашей задачи. На основе полученных данных, мы выяснили, что правительство недополучает 42 066 055,62 МРОТ, (210 330 278 100 руб.) в бюджет Москвы из-за отсутствия штрафов, либо их чрезмерной лояльности. Эти деньги могли бы быть использованы на благие цели, такие как:

- улучшения качества дорожного покрытия
- улучшение системы развязок в пределах Москвы и московской области.
- разработка и построение новых парковочных мест в пределах МКАДа, так как сейчас наблюдается их катастрофическая нехватка
- увеличение зарплаты сотрудникам милиции, работникам службы ТО, с целью уменьшения уровня коррупции
- улучшение и разработка новых, прогрессивных, более экологически чистых двигателей и топлива.

Для решения данной проблемы предполагаются некоторые мероприятия по борьбе с выбросами автотранспорта загрязняющих веществ в атмосферу: Перевод автомобилей на дизельные двигатели. Газ вместо бензина. Электромобиль.

Создаваемые в городах системы движения в режиме зелёной волны, существенно сокращающие число остановок транспорта на

перекрёстках, призваны сократить загрязнения атмосферного воздуха в городах. Большое влияние на качество и количество выбросов примесей оказывает режим двигателя, частичности состояние между массами топлива и воздуха, момент зажигания, качество топлива, отношение поверхности камеры сгорания к её объёму и др. При увеличении отношения массы воздуха и топлива, поступающих в камеру сгорания, сокращаются выбросы оксида углерода и углеводородов, но возрастает выброс оксидов азота.

Несмотря на то, что дизельные двигатели более экономичны, таких веществ, как СО, НпСп, NO_x, они выбрасывают не более, чем бензиновые. Эти двигатели больше выбрасывают дыма – это преимущественно несгоревший углерод. Эти выбросы обладают неприятным запахом, потому что они созданы некоторыми несгоревшими углеводородами. В сочетании с шумами дизельные двигатели не только сильнее загрязняют среду, но и воздействуют на здоровье человека гораздо в большей степени, чем бензиновые.

Количество автомобилей ежегодно увеличивается на 7%. Исходя из нашего опыта, количество автомобилей, проходящих по автотрассе превышает в 6 раз санитарные требования. Это приводит к увеличению оксида углерода в атмосфере, что составляет более 15% от нормы. Выброс свинца в окружающую среду очень велик, что составляет до 5кг/ час и он оседает в районе маги-

страли. Возникает опасность загрязнения почвы, она состоит не только в изменении ее физико-химических свойств. Посторонние вещества, попадая на почву, разрушают сложившиеся связи между отдельными группами почвенного биоценоза. Разрушаются сложившиеся трофические цепи. Всё это, в конечном счёте, сказывается на плодородии почвы и здоровье людей.

Итак, в случае рационального использования средств, полученных из вновь введённых штрафов, общая картина экологического состояния атмосферы города Москвы, качества дорожного покрытия, компетентности работников милиции и работников службы ТО улучшится. Следовательно, по результатам решения нашей задачи общее положение дел в сфере транспорта выйдет на новый уровень.

Список литературы

1. Агаджанян Н. А, Трошкин В. И. - Экология человека / ММП, «Экоцентр круг», 1994.
2. Болбас М.М. - Основы промышленной экологии./ Москва, 1993.
3. Вронский В. А. - Прикладная экология. / 1996.
4. Дубовкин Н. Ф., Фаворский О. Н., Зайцев В. Н. - Газовое топливо на транспорте. / 1991.
5. Криксунов Е. А., Пасечник В. В. - Экология. / «Дрофа», 1995.

Метод сетевого оператора для синтеза систем управления

Дивеев А. И., д.т.н., ВЦ РАН

Приведен анализ структуры данных, польской записи, на предмет ее эффективного использования при поиске математического выражения с помощью генетического программирования. Описывается новая структура данных для машинного представления математических выражений. Данная структура называется сетевой оператор и позволяет представлять математическое выражение в виде ориентированного графа, который в свою очередь представляется в компьютере с помощью целочисленной матрицы.

Метод сетевого оператора [1-4] позволяет построить вычислительную процедуру, которая может обеспечить нахождение оптимального решения в виде математического выражения. Одной из прикладных задач, где это требуется, является задача синтеза оптимального управления. Задача заключается в том, чтобы найти управление, зависящее от состояния объекта. Данная зависимость в общем случае представляет собой математическое выражение, которое описывает функциональную зависимость вектора управления от вектора состояния объекта.

Представим формальную постановку задачи синтеза оптимального управления.

Задана математическая модель объекта управления в виде системы обыкновенных дифференциальных уравнений

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (1)$$

где \mathbf{x} - вектор управления, \mathbf{u} - вектор управления, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $m \leq n$.

Для системы (1) заданы начальные условия

$$\mathbf{x}(0) = \mathbf{x}^0. \quad (2)$$

Заданы ограничения на управление

$$\mathbf{u}^- \leq \mathbf{u} \leq \mathbf{u}^+. \quad (3)$$

Задано терминальное состояние

$$\mathbf{x}(t_f) = \mathbf{x}^f, \quad (4)$$

где t_f - время окончания процесса управления, либо задано, либо ограничено $t_f \leq t^+$.

Задан функционал качества

$$J = \int_0^{t_f} f_0(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (5)$$

Необходимо с учетом ограничений (3) найти управление в виде

$$\mathbf{u} = \mathbf{h}(\mathbf{x}), \quad (6)$$

которое обеспечивает выполнение терминальных условий (4) и минимизирует значение функционала качества (5).

В общем случае предполагаем, что вид функции $\mathbf{h}(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}^m$ заранее не известен, т.е. она не может быть представлена с точностью до значений параметров, например в виде

$$\mathbf{h}(\mathbf{x}) = \mathbf{g}^0(\mathbf{q}, \mathbf{x}), \quad (7)$$

где $\mathbf{g}^0(\mathbf{q}, \mathbf{x})$ - заданная функция, \mathbf{q} - вектор неизвестных параметров.

Для поиска решения задачи используем метод вариации сетевого оператора [1]. Зададим первоначальное базисное решение в виде сетевого оператора. Затем определим вариации этого базисного решения, как вариации ориентированного графа, которые не меняют свойств сетевого оператора. Поиск решения осуществляем на множестве вариаций.

Вариации сетевого оператора не должны изменять конструктивных множеств, с помощью которых он построен, поэтому используем только однотипные сетевые операторы.

Однотипные сетевые операторы - это сетевые операторы, для которых выполняются следующие условия:

- узлы-источники совпадают;
- используют одинаковые конструктивные множества.

Вариация сетевого оператора – это такое изменение сетевого оператора, которое приводит к однотипному сетевому оператору.

Для более качественного поиска решения с помощью эволюционных алгоритмов необходимо использовать малые вариации, которые наименее всего изменяют сетевой оператор. Такими малыми вариациями считаем элементарные вариации.

Вариация сетевого оператора – это такое изменение сетевого оператора, которое приводит к однотипному сетевому оператору.

Элементарная вариация сетевого оператора – это такая вариация, которая не может быть представлена в виде совокупности других вариаций.

В таблице 1 приведены все возможные малые вариации сетевого оператора.

Табл.1.

Элементарные вариации сетевого оператора

Номер вариации	Наименование вариации
0	Изменение унарной операции, связанной с дугой сетевого оператора
1	Изменение бинарной операции, связанной с узлом сетевого оператора
2	Добавление дуги вместе с унарной операцией
3	Удаление дуги, если узел, куда дуга входит, имеет еще входящую дугу
4	Увеличение номеров узлов
5	Уменьшение номеров узлов
6	Добавление узла с бинарной операцией и входящей в узел дуги, связанной с унарной операцией
7	Удаление узла вместе с входящей в него дугой, если узел является узлом-стоком и в него входит только одна дуга.

Теорема 1. Из одного сетевого оператора всегда можно получить однотипный ему сетевой оператор только с помощью элементарных вариаций.

Доказательство. Пусть заданы два однотипных сетевых оператора. Вариация 2 добавления дуги всегда возможна, если существуют в сетевом операторе узлы, между которыми нет дуг. Выполняем вариацию 2 исходного сетевого оператора до тех пор, пока данная вариация возможна. Получаем полный сетевой оператор, однотипный исходному оператору. Далее с помощью вариации 3 удаления дуги из полного сетевого оператора получим сетевой оператор, который структурно эквивалентен желаемому сетевому оператору. После этого с помощью вариаций 0 и 1 изменения унарных и бинарных операций получаем желаемый сетевой оператор.

Для формального описания вариации сетевого оператора введем вектор вариации

$$\mathbf{w} = [w_1 \ w_2 \ w_3 \ w_4]^T \quad (8)$$

где w_1 - номер вариации согласно табл. 1, w_2, w_3, w_4 - целочисленные компоненты, принимающие значения в зависимости от номера вариации.

Для поиска оптимального сетевого оператора, который соответствует оптимальному математическому выражению, используем генетический алгоритм. Генетический алгоритм удобен тем, что он позволяет одновременно искать структуру математического выражения и значения параметров, входящих в него.

Первоначально задаем базисное решение задачи. Это решение может представлять собой любое математическое выражение, которое, по мнению исследователя, наиболее близко к искомому оптимальному выражению.

Базисное решение описываем матрицей сетевого оператора $\Psi^0 = [\psi_{ij}^0]$, $i, j = \overline{1, L}$.

Генерируем упорядоченные множества векторов вариаций

$$W^i = (\mathbf{w}^{i1}, \dots, \mathbf{w}^{il}), i = \overline{1, H}, \quad (9)$$

где $\mathbf{w}^{ij} = [w_1^{ij} w_2^{ij} w_3^{ij} w_4^{ij}]^T$, $i = \overline{1, H}$, $j = \overline{1, l}$, H - количество возможных решений или размерность популяции.

Генерируем битовые строки, определяющие параметрическую часть хромосомы

$$\mathbf{s}^i = [s_1^i \dots s_{p(c+d)}^i]^T, i = \overline{1, H}, \quad (10)$$

где p - число параметров, c - число бит под целую часть числа, d - число бит под дробную часть числа,

Для каждой хромосомы (W^i, \mathbf{s}^i) , $1 \leq i \leq H$, определяем значения функционалов. Для этой цели по структурной части $W^i = (\mathbf{w}^{i1}, \dots, \mathbf{w}^{il})$ хромосомы и базисной матрице Ψ^0 строим матрицу сетевого оператора

$$\Psi^i = \mathbf{w}^{i1} \circ \dots \circ \mathbf{w}^{il} \circ \Psi^0. \quad (11)$$

Параметрическую часть хромосомы $\mathbf{s}^i = [s_1^i \dots s_{p(c+d)}^i]^T$, $1 \leq i \leq H$, переводим в вектор параметров. Для этой цели первоначально битовую строку \mathbf{s}^i из кода Грея переводим в двоичный код

$$\mathbf{b}^i = [b_1^i \dots b_{p(c+d)}^i]^T, \quad (12)$$

$$b_j^i = \begin{cases} s_j^i, & \text{если } (j-1) \bmod (c+d) = 0 \\ s_j^i \oplus b_{j-1}^i, & \text{иначе} \end{cases}, j = \overline{1, p(c+d)}. \quad (13)$$

Затем из двоичного кода получаем вектор параметров $\mathbf{q}^i = [q_1^i \dots q_p^i]^T$,

$$q_k^i = \sum_{j=1}^{c+d} 2^{c-j} b_{j+(k-1)(c+d)}^i, k = \overline{1, p}. \quad (14)$$

Для каждой полученной матрицы сетевого оператора Ψ^i и вектора параметров \mathbf{q}^i определяем математическое выражение $\mathbf{g}^i(\mathbf{x}, \mathbf{q}^i)$ и оцениваем его с помощью функции приспособленности

$$f_i^0 = F_0(W^i, \mathbf{q}^i). \quad (16)$$

Функция приспособленности – это целевая функция оптимизационной задачи или функция, построенная на основе целевой функции (5) и терминальных условий (4).

Среди всех возможных решений в популяции находим решение, которое дает наименьшее значение функции приспособленности

$$f_{i_-}^0 = \min_i \{F_0(W^i, \mathbf{q}^i), i = \overline{1, H}\}, \quad (17)$$

где i_- - номер наилучшего решения или хромосомы в текущей популяции.

Выбираем случайно два решения $(W^{i_1}, \mathbf{s}^{i_1})$, $(W^{i_2}, \mathbf{s}^{i_2})$ и определяем возможность их скрещивания. Для этой цели вычисляем вероятность скрещивания

$$p_c = \max \left\{ \frac{f_{i_-}^0}{f_{i_1}^0}, \frac{f_{i_-}^0}{f_{i_2}^0} \right\}. \quad (17)$$

Генерируем случайную величину $\xi \in [0,1)$. Если выполняются условия $\xi < p_c$, то осуществляем скрещивание

Если условие скрещивания выполняется, то выбираем случайно точки скрещивания для структурной и параметрической частей хромосомы k_s и k_p .

Осуществляем построение новых хромосом-потомков (W^{H+1}, s^{H+1}) , (W^{H+2}, s^{H+2}) , (W^{H+2}, s^{H+3}) , (W^{H+4}, s^{H+4}) . Всего получаем четыре потомка, для двух потомков скрещиваем только параметрические части, оставляя неизменными структурные, а для двух других потомков скрещиваем и структурные и параметрические части

$$W^{H+1} = W^{i_1}, \quad W^{H+2} = W^{i_2}, \quad s^{H+1} = \left[s_1^{i_1} \dots s_{k_p-1}^{i_1} s_{k_p}^{i_2} \dots s_{p(c+d)}^{i_2} \right]^T,$$

$$s^{H+2} = \left[s_1^{i_2} \dots s_{k_p-1}^{i_2} s_{k_p}^{i_1} \dots s_{p(c+d)}^{i_1} \right]^T, \quad W^{H+3} = (w^{i_1,1}, \dots, w^{i_1, k_s-1}, w^{i_2, k_s}, \dots, w^{i_2, l}),$$

$$s^{H+3} = \left[s_1^{i_1} \dots s_{k_p-1}^{i_1} s_{k_p}^{i_2} \dots s_{p(c+d)}^{i_2} \right]^T, \quad s^{H+4} = \left[s_1^{i_2} \dots s_{k_p-1}^{i_2} s_{k_p}^{i_1} \dots s_{p(c+d)}^{i_1} \right]^T,$$

Использование метода сетевого оператора позволило решить много задач синтеза оптимального управления для различных прикладных объектов. В частности, решены задачи синтеза системы стабилизации угловым движением спутника [1], задача синтеза оптимальным маневром летательного аппарата [2], задача синтеза оптимального управления спуском космического аппарата [3], задача синтеза управления подъемом метеорологической ракетой [4] и др.

Работа выполнена по гранту РФФИ № 08-08-00248-а.

Литература

1. Diveyev A.I., Sofronova E.A. Application of network operator method for synthesis of optimal structure and parameters of automatic control system// Proceedings of 17-th IFAC World Congress, Seoul, 05.07.2008 – 12.07.2008. P. 6106 – 6113.
2. Diveev A.I., Sofronova E.A. Method of network operator for synthesis of optimal control system// Al Taknie (Libya) 2008, no 6. P. 130-143.
3. Дивеев А.И., Шмалько Е.Ю. Многокритериальный структурно-параметрический синтез системы управления спуском космического аппарата на основе метода сетевого оператора// Вестник Российского университета дружбы народов. Серия инженерные исследования (информационные технологии и управление). 2008, № 4. С. 86 - 93.
4. Дивеев А.И., Северцев Н.А., Софронова Е.А. Синтез системы управления метеорологической ракетой методом генетического программирования// Проблемы машиностроения и надежности машин. 2008, № 5. С. 104 - 108.

О безопасности функционирования динамической системы на основе технической устойчивости

Дружинина О. В., д. ф.-м. н., проф.,

Вычислительный центр им. А.А. Дородницына РАН

Северцев Н. А., д. т. н., проф.,

Вычислительный центр им. А.А. Дородницына РАН

Масина О. Н., к. ф.-м. н., доцент

Елецкий государственный университет им. И. А. Бунина

Дан сравнительный анализ различных типов устойчивости на конечном интервале времени. Предложен способ оценки безопасности функционирования систем с использованием понятия технической устойчивости.

В настоящее время развитие средств производства и транспорта характеризуется созданием все более сложных технических систем и технологических процессов. Как при эксплуатации современных технических систем, так и при внедрении новых технологических процессов, в связи с увеличением числа составляющих их элементов и усложнением взаимосвязей между ними, увеличивается число отказов и, соответственно, число технических и техногенных катастроф. Кроме того, ситуация, сложившаяся к настоящему времени, характеризуется и значительным износом оборудования. Часто это приводит к авариям, для ликвидации последствий которых привлекается достаточно большое количество ресурсов.

В связи с указанными обстоятельствами возникает, с одной стороны, необходимость создания новых математических моделей, описывающих динамические процессы, а с другой стороны, необходимость разработки методов, позволяющих оценивать безопасность функционирования динамических систем. Так, на железнодорожном транспорте в связи с проектированием и внедрением скоростных и высокоскоростных составов актуальными задачами являются изучение качественного поведения и устойчивости математических динамических моделей и учет различных типов возмущений, возникающих при взаимодей-

ствии колеса и рельса, а также вопросы динамической безопасности и управления движением.

Одним из показателей качества функционирования динамической системы является техническая (практическая) устойчивость, изучаемая на фиксированном промежутке времени [1–8]. Во многих прикладных задачах важно изучение поведения системы на фиксированном промежутке времени (см., например, [3, 4, 6–8]).

Изучение технической устойчивости систем оказывается полезным в тех случаях, когда другие понятия устойчивости (например, устойчивость по Ляпунову) не соответствуют постановке задачи в том смысле, что в ряде случаев не представляется возможным рассматривать движение на бесконечных промежутках времени или особенности структуры фазовых множеств, требуют использования понятия технической устойчивости.

Главная задача теории технической устойчивости состоит в изучении движения при заданных оценках областей начальных и последующих возмущений в течение фиксированного промежутка времени. Техническая устойчивость является не только качественным, но и количественным показателем динамической системы. Этот показатель дает возможность оценить качество самой динамической системы и ответить на вопрос, какая из двух изучаемых систем является

более устойчивой и, следовательно, более безопасной системой.

Важной задачей в исследовании качества функционирования динамической системы является получение количественных оценок, обеспечивающих наличие технической устойчивости, причем технология самого исследования обладает следующими особенностями: движение изучаемой системы оценивается на некотором заранее заданном промежутке времени при учете конечных возмущающих сил, причем начальные и последующие возмущения являются конечными величинами; исследование, как правило, требует привлечения различных типов функций Ляпунова [3], являющихся эффективным инструментом изучения устойчивости.

Известно большое число различных определенных понятия технической (практической) устойчивости, однако при всем их разнообразии указанные определения имеют одни и те же предпосылки. А именно, в каждой постановке задачи о технической устойчивости:

1) рассматривается дифференциальное уравнение $\dot{x} = g(x)$ или $\dot{x} = g(t, x)$, где $x \in R^n$; 2) вводится определенное множество $M_0(t) \subset R^n$ начальных возмущений $x_{i0} = x_i(t_0)$ ($i = 1, 2, \dots, n$) и рассматриваются траектории, исходящие из точек $x_0 \in M_0(t_0)$; 3) задается определенный интервал $I \subset I_0$ значений времени t ; 4) вводится определенное множество $M_1(t)$ допустимых значений возмущений $x(t)$ на I .

Понятие технической устойчивости принимает конкретное содержание в зависимости от выбора множеств $M_0(t)$, $M_1(t)$ на I и этот выбор имеет существенное значение в отличие от постановки задачи об устойчивости по Ляпунову.

Рассмотрим динамическую систему, описываемую дифференциальным уравнением

$$\frac{dx}{dt} = g(t, x), \quad (1)$$

где $g(t, x)$ – заданная векторная функция от входящих аргументов, где $t \in I := [t_0, t_0 + \tau)$, $\tau < +\infty$, $x \in R^n$ – вектор состояния. Предполагается, что уравнение удовлетворяет условиям существования, единственности и непрерывной зависимости решений.

В n -мерном пространстве R^n фазовых координат задаются два фазовых множества $M_0(t)$ и $M_1(t)$ системы (1), вообще говоря, изменяемые во времени t , причем первое множество $M_0(t)$ вложено во второе $M_1(t)$, т.е. $M_0(t) \subseteq M_1(t) \quad \forall t \in I$, где $I := [t_0, t_0 + \tau)$, $\tau < +\infty$.

Множества $M_0(t)$ и $M_1(t)$ предполагаются ограниченными, связными, открытыми подмножествами в R^n . Обозначим через $\bar{M}_i(t)$ и $\partial M_i(t)$ замыкание и границу множества $M_i(t)$ ($i = 0, 1$).

Фазовое множество $M_0(t)$ называется *технически устойчивым на временном конечном промежутке I* относительно множества $M_1(t)$, если все полуинтегральные кривые, начинающиеся в момент времени $t = t_0$ во множестве $M_0(t_0)$ при всех значениях t из заданного промежутка I содержатся в множестве $M_1(t)$. Если же хотя бы одна полуинтегральная кривая, начинающаяся при $t = t_0$ во множестве $M_0(t_0)$ выходит в какой-то момент t^* из множества $M_1(t)$, то фазовое множество $M_0(t)$ называется *технически неустойчивым на I* относительно фазового множества $M_1(t)$.

В зависимости от способа задания множеств $M_0(t)$, $M_1(t)$ определение технической устойчивости на временном интервале I множества $M_0(t)$ относительно $M_1(t)$ принимает конкретное содержание. Выбор множеств $M_0(t)$ и $M_1(t)$ имеет существенное значение, так как множества $M_i(t)$ ($i = 0, 1$), выбранные в виде n -мерного шара и в виде n -мерного параллелепипеда, приводят к различным понятиям технической устойчивости.

Рассмотрим динамические системы $\phi^1: X \rightarrow X$ и $\phi^2: X \rightarrow X$ на совокупностях (M_0^1, M_1^1, I^1) и (M_0^2, M_1^2, I^2) , где X – фазовое пространство (например, пространство координат), I^1, I^2 – фиксированные интервалы времени для ϕ^1 и ϕ^2 соответственно, причем первая совокупность определяет техническую устойчивость динамической системы ϕ^1 , а вторая совокупность – техническую устойчивость динамической системы ϕ^2 . Будем предполагать, что множества M_i^j ($i = 0, 1; j = 1, 2$) вложены друг в друга в теоретико-множественном смысле. Обозначим через M_{\min}^j и M_{\max}^j наименьшее и наибольшее из множеств M_i^j соответственно. Тогда динамическая система ϕ^1 называется *более устойчивой*, чем система ϕ^2 , если множество M_{\min}^1 является множеством начальных условий, а множество M_{\max}^2 – множеством последующих условий. В этом случае также будем называть систему ϕ^1 более безопасной, чем система ϕ^2 .

Определение безопасности функционирования динамической системы $\phi: X \rightarrow X$ на определяющих техническую устойчивость двух совокупностей (M_0^j, M_1^j, I^j) ($j = 1, 2$) может быть сформулировано следующим образом: динамическая система $\phi: X \rightarrow X$ называется *устойчивой (безопасной)*, если она устойчива на совокупности (M_{\min}, M_{\max}, I) , причем в качестве I можно выбирать один из интервалов I^1, I^2 .

Введенные определения позволяют дать оценку безопасности функционирования одной динамической системы по сравнению с другой. Отметим, что установленные в [6–8] достаточные условия технической устойчивости движения могут служить основой методики для анализа эффективности функционирования динамических систем.

В настоящее время метод функций Ляпунова стал основным методом исследования технической устойчивости одного фазового множества относительно другого фазового множества. Метод функций Ляпунова состоит в том, что задача исследования устойчивости движения сводится к отысканию некоторых функций переменных t, x_1, x_2, \dots, x_n (функции Ляпунова), полные производные которых в силу уравнений возмущенного движения обладают некоторыми определенными свойствами. Функции Ляпунова представляют собой вещественные функции вещественных переменных x_1, x_2, \dots, x_n или переменных t, x_1, x_2, \dots, x_n , определенные соответственно в областях $D = \{x : \|x\| < d\}$, где d – положительное число или символ ∞ , и $D_t = I_0 \times D$, где $I_0 \subset [0 \leq t < \infty)$ – некоторый интервал (замкнутый, открытый или полуоткрытый) временной оси.

Приведенные в [6–8] результаты позволяют провести исследования технической устойчивости движения динамических систем в тех случаях, когда понятие устойчивости по Ляпунову для данной транспортной системы не соответствует постановке задачи и когда модель движения задается обыкновенными неавтономными дифференциальными уравнениями первого и второго порядков. Полученные результаты могут быть использованы для решения задач устойчивости и безопасности функционирования транспортных и инженерных систем.

Одним из направлений дальнейших исследований является получение оценок безопасности функционирования динамических систем, описываемых дифференциальными уравнениями второго порядка, а также динамических систем с различными типами нелинейностей, которые не рассмотрены в настоящей работе. Интерес представляет также распространение изложенных методов на исследование устойчивости и безопасности функционирования систем, задаваемых линейными и нелинейными матрично-

векторными дифференциальными уравнениями, встречающимися в теории автоматического регулирования, динамике подвижного состава железнодорожного транспорта, теории управления, теории стабилизации движения и других областях.

Литература

1. *Четаев Н.Г.* Об одной мысли Пуанкаре // Сб. научных трудов Казанского авиац. ин-та. 1935. № 3. С. 3–6.
2. *Карачаров К.А., Пилютик А.Г.* Введение в техническую теорию устойчивости движения. М.: Физматгиз, 1962.
3. *Зубов В.И.* Математические методы исследования систем автоматического регулирования. Ленинград: Машиностроение, 1974.
4. *Мартынюк А.А.* Практическая устойчивость движения. Киев: Наукова думка, 1983.
5. *Абгарян К.А.* Введение в теорию устойчивости движения на конечном интервале времени. М.: Наука, 1992.
6. *Захарова М.В., Дружинина О.В., Мулкиджан Т.С.* Оценка безопасности функционирования динамической системы на основе технической устойчивости // Научно-технические технологии. 2007. Т. 8. № 10. С. 32–40.
7. *Шестаков А.А., Дружинина О.В., Захарова М.В.* О технической устойчивости фазового множества относительно другого фазового множества в нестационарной динамической системе // Методы исследования технической устойчивости и качественных свойств систем железнодорожного транспорта. Межвуз. сб. научн. трудов. М.: РГОТУПС, 2003. С. 60–62.
8. *Дружинина О.В., Масина О.Н.* О методе оценки безопасности функционирования динамической системы на основе технической устойчивости // Труды 16 Международной конференции «Проблемы управления безопасностью сложных систем». М.: ИПУ РАН, 2008. С. 302–306.

Исследование управляемых динамических процессов методом предельных уравнений

Дружинина О. В., д.ф.-м.н., профессор

Вычислительный центр им. А.А. Дородницына РАН

Климачкова Т. С., аспирант

Московский гос. ун-т путей сообщения

Изучены предельные свойства обобщенных динамических процессов с управлением, задаваемых многомерными нелинейными дифференциальными уравнениями.

Будем рассматривать неавтономное управляемое векторное уравнение

$$\dot{x} = f(t, x, u), \quad x \in R^n, \quad u \in U, \quad (1)$$

и будем предполагать, что выполнены следующие условия P_1 – P_3 .

P_1 . Отображение f непрерывно по совокупности (x, u) и измеримо по t , а семейство U всех допустимых управлений состоит из всех измеримых по Лебегу функций $u: R \rightarrow M$, где M – локально компактное метрическое пространство с метрикой d .

P_2 . Для каждого допустимого управления $u(t) \in U$ и каждой пары $(t_0, x_0) \in R \times R^n$ начальная задача

$$\dot{x} = f(t, x, u(t)), \quad x(t_0) = x_0, \quad (2)$$

имеет единственное решение $x(t, t_0, x_0, u)$, определенное для всех $t \in R^+$.

P_3 . Для любого ограниченного множества $B \subset R^n$ существует постоянная $k = k(B)$ такая, что $\|f(t, x, u) - f(t, y, u)\| \leq k\|x - y\| \quad \forall x, y \in B, \quad \forall t \in R, \quad \forall u \in U$.

Отметим, что изучение динамических процессов и их связей с неавтономными дифференциальными уравнениями проводилось в [1–4]. Предельные свойства динамических процессов с управлением и без управления изучены в [5–7].

Динамическим (соответственно полудинамическим) процессом на R^n называется отображение $S: R \times R \times R^n \rightarrow R^n$ (соответственно $S: R^+ \times R \times R^n \rightarrow R^n$) такое, что

- 1) S непрерывно,
- 2) $S(0; t_0, x_0) = x_0$,
- 3) $S(t+\tau; t_0, x_0) = S(t; t+\tau; S(\tau; t_0, x_0)) \quad \forall t, \tau, t_0, x_0$.

Динамический (полудинамический) процесс S называется динамической (полудинамической) системой, если отображение S не зависит от t_0 , т.е.

$S = S(t, x_0)$. Динамический (полудинамический) процесс будем называть D -процессом (D^+ -процессом).

Динамический процесс S описывает изменения, происходящие с течением времени в некоторой материальной системе, при этом пара (t_0, x_0) задает начальное временное условие t_0 и начальное положение x_0 . Отображение $t \rightarrow S(t; t_0, x_0)$ называется движением D (D^+ -процесса). Начальная задача (2) означает, что при $t=0$ движение $t \rightarrow S(t; t_0, x_0)$ начинается в точке x_0 .

Если

- 1) u – допустимое управление,
- 2) $x(t; t_0, x_0, u)$ – решение задачи (2), то функция $S(t; t_0, x_0) := x(t+t_0; t_0, x_0, u)$

является динамическим процессом на R^n [1, 2].

Последовательность $S_k, k = 1, 2, 3, \dots$, динамических процессов называется сходящейся к динамическому процессу S , если

$$S_k(t, t_0, x_0) \xrightarrow{R^n} S(t, t_0, x_0) \quad \text{при } k \rightarrow \infty$$

равномерно на компактных подмножествах из $R \times R \times R^n$.

Обозначим через Π совокупность всех D -процессов S на R^n .

Динамический процесс S называется обобщенным динамическим процессом (относительно (1)), если существует последовательность u_k допустимых управлений из U такая, что последовательность $S_k = S(t; t_0, x_0, u_k)$ в пространстве Π сходится к S .

Семейство динамических процессов, порожденных допустимыми управлениями, называется компактным, если каждая подпоследовательность

ность семейства сходится к (обобщенному) динамическому процессу.

Будем говорить, что функция f удовлетворяет условию P_3 , если:

а) она локально ограничена и локально липшицева по x , т.е. для каждого ограниченного множества $B \subset R \times R^n$ существует постоянная $k > 0$ такая, что

$$\|f(t; x, u)\| \leq K \text{ и } \|f(t; x, u) - f(t; y, u)\| \leq K\|x - y\|, \\ \forall (t, x) \in B, (t, y) \in B;$$

б) множество M компактно, и семейство U состоит из всех измеримых функций $u: R \rightarrow M$.

Рассмотрим связь между управлениями в (1) и порождаемыми уравнениями (1) динамическими процессами. Пусть \mathcal{M} – метрическое пространство вероятностных мер на M , наделенное метрикой, порожденной слабой сходимостью, т.е. $\mu_k \rightarrow \mu$,

если $\int_M g d\mu_k$ сходится к $\int_M g d\mu$ для любого непрерывного отображения $g: M \rightarrow R$.

Вложим U в такое топологическое пространство \mathfrak{R} , в котором U всюду плотно. В частности, можно вложить U в пространство, топологически двойственное $L^1(R, C(R, M))$, отождествляя каждое $r \in U$ с непрерывным линейным функционалом l_r на $L^1(R, C(R, M))$, определенным условием

$$l_r(\varphi) ::= \int \varphi(t; r(t)) d\mu_t \quad (\varphi \in L^1(R, C(R, M))).$$

Пространство \mathfrak{R} называется семейством допустимых релаксированных управлений. Известно [8], что \mathfrak{R} есть метрическое, компактное пространство, и сходимость последовательности r_k к элементу ν в \mathfrak{R} эквивалентна соотношению

$$\int_F \int_M g(t, u) dr_k dt \rightarrow \int_F \int_M g(t, u) dr dt$$

для всякого ограниченного интервала F и числовой непрерывной по u и измеримой по t функции g такой, что $m(t) = \max \{\|g(t, u)\| : u \in M\}$ интегрируема на всем интервале F .

Допустимое управление $u \in U$ рассматривается как элемент \mathfrak{R} , если считать $u(t)$ мерой Дирака, сконцентрированной в точке $u(t) \in M$. Известно [8], что множество допустимых управлений U плотно в \mathfrak{R} , и каждое релаксированное управление порождает обобщенный динамический процесс.

Рассмотрим далее автономный управляемый динамический процесс, описываемый дифференциальным уравнением вида

$$\dot{x} = f(x, u). \quad (3)$$

Для конкретной функции $u(t)$ соответствующее уравнение $\dot{x} = f(x(t), u(t))$ является автономным.

Допустимым решением $x = x(t)$ называется решение уравнения (3) для допустимого управления $u = u(t) \in U$, т.е. $\dot{x} = f(x(t), u(t))$.

Имеет место следующая

Теорема 1. Пусть

- 1) U инвариантно относительно сдвигов,
- 2) дифференциальное уравнение (3) динамически компактно,
- 3) $\Omega(x)$ – положительное предельное множество допустимого решения $x = x(t)$ дифференциального уравнения (3). Тогда для каждого $z \in \Omega(x)$ существует обобщенный динамический процесс S такой, что $S(t; 0, z) \in \Omega(x) \quad \forall t \in R$.

Теорема 2. Пусть

- 1) множество U инвариантно относительно сдвигов,
- 2) $u = u(t)$ – допустимое управление,
- 3) $\Omega(x)$ – компактное множество, где $x = x(t)$ – решение уравнения (3). Тогда предельное множество $\Omega(x)$ обладает следующим свойством: если S – обобщенный динамический процесс, являющийся пределом последовательности $S(u_j^t)$ для некоторой последовательности $\{t_j\}$, $t_j \rightarrow \infty$, то существует точка $z \in \Omega(x)$ такая, что $S(t; 0, z) \in \Omega(x) \quad \forall t \in R$.

Сдвигом f^τ функции f на величину τ называется функция $f^\tau(t, x, u) = f(t + \tau, x, u)$. Управляемое уравнение

$$\dot{x} = g(t, x, u), \quad (4)$$

называется предельным уравнением по отношению к уравнению (2), если существует последовательность $\{\tau_j\}$, $\tau_j \rightarrow \infty$ такая, что для любых t, x, u имеет место

$$\int_0^t g(\mu, x, u) d\mu = \lim_{j \rightarrow \infty} \int_0^t f^{\tau_j}(\mu, x, u) d\mu.$$

Последовательность f^{τ_j} называется сходящейся к g .

Теорема 3. Семейство предельных уравнений замкнуто относительно сдвигов, т.е. если уравнение (4) есть предельное уравнение, то уравнение $\dot{x} = g^\tau(t, x, u)$ есть предельное уравнение для любого $\tau \in R$.

Теорема 4. Для каждой последовательности $\{\tau_j\}$, $\tau_j \rightarrow \infty$, существует подпоследовательность τ_j для которой последовательность f^{τ_j} сходится.

Пусть $r \in \mathfrak{R}$, где \mathfrak{R} – семейство допустимых релаксированных управлений.

Теорема 5. Пусть

- 1) $S(t, t_0, x_0; r, g)$ – динамический процесс, порожденный уравнением (1) с управлением r ,
- 2) r^τ и S^τ – сдвиги на величину τ функций r и S соответственно, т.е.

$$r^\tau(t) ::= r(t + \tau) \text{ и } S^\tau(t, t_0, x_0) ::= S(t, t_0 + \tau, x_0).$$

Тогда имеет место равенство
 $S^r(t, t_0, x_0; r, g) = S(t, t_0, x_0; r^r, g^r)$.

Теорема 6. Пусть последовательность u^i сходится к $r \in \mathcal{Y}$ и последовательность f^i сходится к g . Тогда последовательность $S^i(u, f)$ сходится к $S(r, g)$ при $i \rightarrow \infty$.

Теорема 7. Пусть

- 1) u – допустимое управление,
- 2) $x = x(t)$ – решение задачи (2). Тогда для каждого $z \in \Omega(x)$ существуют предельное уравнение (4) и релаксированное управление r такие, что $S(t, 0, z; r, g) \in \Omega(x) \quad \forall t \in R^+$, где $\Omega(x)$ – положительное предельное множество точки x .

Теорема 7 позволяет рассматривать класс уравнений $\dot{x} = g(t, x, r(t))$ для предельных уравнений $\dot{x} = g$ и всех релаксированных управлений r , относительно которых множество $\Omega(x)$ несмещенное.

Полученные результаты являются продолжением работ [9, 10].

Литература

1. *Dafermos C.M.* An invariance principle for compact processes // J. of Differential Equations. 1971. V. 9. P. 239–252.
2. *Dafermos C.M.* Applications of the invariance principle for compact processes // J. of Differential Equations. 1971. V. 9. P. 291–299.
3. *Sell G.R.* Nonautonomous differential equations and topological dynamics // Trans. AMS. 1967. V. 127. № 3. P. 241–283.
4. *Artstein Z.* Stability, observability and invariance // J. of Differential Equations. 1982. V. 44. P. 224–248.
5. *Шестаков А.А.* Прямой метод Ляпунова как метод локализации функциями Ляпунова предельных множеств неавтономных динамических процессов // Функции Ляпунова и их применения. Новосибирск: Наука, 1987. С. 14–48.
6. *Шестаков А.А.* Обобщенный прямой метод Ляпунова для систем с распределенными параметрами. М.: УРСС, 2007.
7. *Баймаков И.Г.* Об управлении динамической системой // Проблемы динамики подвижного состава и устойчивости движения динамических систем. Сб. науч. трудов. М.: ВЗИИТ, 1990. С. 118–120.
8. *Варга Дж.* Оптимальное управление дифференциальными и функциональными уравнениями. М.: Наука, 1977.
9. *Дружинина О.В., Масина О.Н., Климачкова Т.С.* О предельных свойствах и инвариантности управляемых динамических процессов // Вопросы теории безопасности и устойчивости систем. Вып. 10. М.: ВЦ РАН, 2008. С. 55 – 63.
10. *Дружинина О.В., Климачкова Т.С.* О методе предельных уравнений исследования управляемых динамических процессов. Труды института системного анализа РАН. Динамика неоднородных систем, 2008. Т. 32 (1). С. 54 – 57.

Об устойчивости и управляемости транспортных систем

Масина О. Н., к. ф.-м. н., доцент

Елецкий государственный университет им. И. А. Бунина

Дружинина О. В., д. ф.-м. н., профессор

Вычислительный центр им. А.А. Дородницына РАН

Иванова Ю. А., аспирант

Вычислительный центр им. А.А. Дородницына РАН

Игонина Е. В., аспирант

Елецкий государственный университет им. И.А. Бунина

Рассмотрены вопросы управления движением некоторых транспортных систем, описываемых дифференциальными уравнениями в контингенциях.

Решение ряда приоритетных задач управления сложными техническими объектами и технологическими процессами, а также задач разработки автоматизированных систем управления связано с изучением вопросов устойчивости и управляемости транспортных систем, играющих важную роль в развитии динамики управляемых систем транспорта.

Вопросам динамики управляемых систем посвящены многочисленные работы (см., например, [1–9]). В настоящей статье рассмотрены вопросы управления движением некоторых транспортных систем, описываемых дифференциальными уравнениями в контингенциях, что позволяет провести анализ устойчивости номинального движения, определяемого критерием оптимальности. Многозначность в правых частях дифференциальных уравнений возникает ввиду сопротивления разреженной среды, которое в первом приближении не учитывается.

Пусть управляемый объект перемещается в вертикальной плоскости $хоу$ из точки $O(0; 0)$ в точку $L(l; 0)$ с промежуточным достижением высоты h . При этом перемещение объекта на плоскости $хоу$ разбивается на два этапа. На первом этапе $0 \leq t \leq t_1$ объект движется из начальной точки $O(0; 0)$ с постоянным ускорением вверх, и этап заканчивается достижением точки $H(t_1; h)$. На втором этапе $t_1 \leq t \leq t_2$ он движется из точки H с постоянным ускорением вниз до точки $L(l; 0)$ оси $ох$. Предполагается, что во время движения масса объекта m не меняется.

На первом этапе движение происходит с постоянной векторной тягой (p, q) ; на втором этапе – с постоянной тягой $(-r, s)$, причем, в отличие от исследований других авторов [3, 4], значения положительных параметров p, q, r, s выбираются из интервалов $(p_1, p_2), (q_1, q_2), (r_1, r_2), (s_1, s_2)$ соответственно. На объект действует также сила тяготения с постоянным ускорением g , а сопротивлением среды пренебрегаем, как будто движение проводится в вакууме.

Движение будет допустимым, если координаты x, y объекта удовлетворяют условиям: $x(t)$ растет от 0 до l , $y(t)$ возрастает на первом этапе до высоты h и $y(t)$ убывает до нуля на втором этапе. При этом в момент приземления t_2 , то есть когда $y(t_2)=0$, получаем $x(t_2)=l$.

Движение будет оптимальным, если на перемещение объекта тратится меньше топлива. Критерием качества является условие:

$$\forall p \in (p_1, p_2), \forall q \in (q_1, q_2), \forall r \in (r_1, r_2), \forall s \in (s_1, s_2), \int_0^{t_1} (p+q)dt + \int_{t_1}^{t_2} (r+s)dt \rightarrow \min. \quad (1)$$

Задача оптимального управления заключается в нахождении параметров p, q, r, s и значений t_1, t_2 , чтобы движение объекта из начальной точки $O(0; 0)$ в конечную $L(l; 0)$ с промежуточным достижением высоты h осуществлялось с минимальным расходом топлива.

Предлагается следующий алгоритм решения задачи. Первым шагом является нахождение параметров p , q , r и s . Выберем сначала однозначную реализацию, состоящую из единственных значений этих параметров. Тогда дифференциальные включения

$$\begin{aligned} mx'' &\in p, \\ my'' &\in q - mg, \end{aligned}$$

и

$$\begin{aligned} mx'' &\in -r, \\ my'' &\in s - mg, \end{aligned}$$

описывающие движения объекта на первом и втором этапах, преобразуются в системы дифференциальных уравнений второго порядка с постоянными правыми частями:

$$\begin{aligned} mx'' &= p, \\ my'' &= q - mg, \end{aligned}$$

на участке разгона и

$$\begin{aligned} mx'' &= -r, \\ my'' &= s - mg, \end{aligned}$$

на участке торможения.

Для упрощения выкладок проведем замену: $P_1 := p/m$, $Q_1 := (q - mg)/m$, $R_1 := r/m$ и $S_1 := (s - mg)/m$. Тогда на первом этапе $0 \leq t \leq t_1$ движение объекта будет описываться системой дифференциальных уравнений второго порядка

$$\begin{aligned} x'' &= P_1, \\ y'' &= Q_1, \end{aligned} \quad (2)$$

с условиями $x(0) = y(0) = x'(0) = y'(0) = 0$, $y(t_1) = h$.

На втором этапе $t_1 \leq t \leq t_2$ движение объекта описывается системой дифференциальных уравнений второго порядка с соответствующими начальными условиями в точке $t = t_1$:

$$\begin{aligned} x'' &= -R_1, \quad x(t_1) = \frac{P_1 t_1^2}{2}, \quad x'(t_1) = P_1 t_1, \\ y'' &= -S_1, \quad y(t_1) = \frac{Q_1 t_1^2}{2} = h, \quad y'(t_1) = Q_1 t_1, \quad t_1 \leq t \leq t_2. \end{aligned} \quad (3)$$

Решая (2) и (3) и полагая $\tau = t_2/t_1$, получим систему:

$$\begin{aligned} P_1(2\tau - 1) - R_1(\tau - 1)^2 &= \frac{2l}{t_1^2}, \\ P_1 &= R_1(\tau - 1), \\ -S_1(\tau - 1)^2 + Q_1(2\tau - 1) &= 0, \\ Q_1 &= \frac{2h}{t_1^2}, \end{aligned}$$

решение которой имеет вид:

$$P_1 = \frac{2l}{t_1^2 \tau}, \quad R_1 = \frac{2l}{t_1^2 \tau (\tau - 1)}, \quad S_1 = \frac{2(2\tau - 1)h}{t_1^2 (\tau - 1)^2}, \quad Q_1 = \frac{2h}{t_1^2}.$$

После нахождения выражений для P_1 , R_1 , S_1 и Q_1 , переходим ко второму шагу решения задачи – вычислению параметров τ и t_1 . Для этого воспользуемся условием оптимальности (1), где

$$F(t_1, \tau) := m \left(\frac{4l}{t_1 \tau} + \frac{6h}{t_1} + \frac{2h}{t_1 (\tau - 1)} + g t_1 \tau \right).$$

Для вычисления критических точек решаем систему:

$$\begin{aligned}\frac{4l}{\tau} + 6h - t_1^2 \tau g + \frac{2h}{\tau - 1} &= 0, \\ \frac{4l}{\tau} - t_1^2 \tau g + \frac{2h\tau}{(\tau - 1)^2} &= 0,\end{aligned}$$

откуда ввиду $\tau > 1$ получим $\tau = 1 + \frac{1}{\sqrt{3}}$.

Тогда

$$t_1 = \sqrt{\frac{4l(\tau - 1)^2 + 2h\tau^2}{\tau^2(\tau - 1)^2 g}} \quad \text{и} \quad t_2 = \tau \sqrt{\frac{4l(\tau - 1)^2 + 2h\tau^2}{\tau^2(\tau - 1)^2 g}}.$$

Зная параметры τ и t_1 , найдем значения p , q , r и s :

$$p = \frac{2l\tau(\tau - 1)^2 gm}{4l(\tau - 1)^2 + 2h\tau^2}; \quad q = \frac{2h\tau^2(\tau - 1)^2 gm}{4l(\tau - 1)^2 + 2h\tau^2} + mg; \quad r = \frac{2l\tau(\tau - 1)gm}{4l(\tau - 1)^2 + 2h\tau^2}; \quad s = \frac{2h\tau^2(2\tau - 1)gm}{4l(\tau - 1)^2 + 2h\tau^2} + mg.$$

В частности, проведенные вычисления в пакете Mathematica 4.1 показали прямолинейную зависимость силы тяги от h , l и массы объекта [8].

Так как параметры p , q , r , s многозначные и выбираются из соответствующих интервалов (p_1, p_2) , (q_1, q_2) , (r_1, r_2) , (s_1, s_2) , то будет отклонение от оптимального движения. Поэтому последним этапом в решении задачи является коррекция движения с помощью датчиков для возвращения на оптимальный маршрут.

После конкретизации условий рассмотренная система может быть дополнена: учесть сопротивление среды, учесть изменение массы носителя, учесть возможное перемещение цели и сопротивление противника, если он имеется. Тогда систему перемещения тела из начальной точки в конечную с промежуточным достижением заданной высоты можно использовать при перемещении груза с помощью летательного аппарата в условиях, когда достижение конечной точки по малой окрестности поверхности земли невозможно. В отличие от рассмотренной системы, когда все решения являются элементарными функциями, в усложненной системе управление должно рассчитываться численными методами и обязательно потребуются разработка проверки устойчивости.

Рассмотренную выше задачу можно использовать для решения более сложной задачи, в которой перемещаемому объекту необходимо поразить движущуюся цель, перемещающуюся в плоскости xOz со скоростью v в произвольном направлении.

Предлагается следующий алгоритм решения задачи.

Шаг 1. На первом этапе движение объекта происходит из точки $O(0, 0, 0)$ с постоянным ускорением вверх и вперед до точки $H(0, h, 0)$, как и в первой задаче. За время движения t_1 цель перемещается в точку круга (круга положений).

Шаг 2. Из всех возможных траекторий движения объекта выбираем ту, которая ведет в точку нахождения цели. За оставшееся время $t = t_2 - t_1$, как и в первой задаче, объект достигнет плоскости xOz , направляясь в центр круга положений.

Движение по этой траектории будем проводить за время $t/2$, при этом устанавливаем новый круг положений, в котором окажется цель. Выбираем ту траекторию, которая ведет в центр круга положений. Если радиус этого круга не больше радиуса поражения r , то оставшееся время $t/2$ тратим на движение в центр этого круга с поражением цели.

Шаг 3. Если радиус этого круга больше радиуса поражения, то повторяем шаг 2, где время движения, за которое объект проходит половину маршрута в центр круга положений, уменьшается вдвое. Если радиус этого круга не больше радиуса поражения r , то оставшееся время $t/4$ тратим на движение в центр этого круга с поражением цели. Если это не так, то повторяем второй шаг с уменьшенным вдвое временем и так далее.

Так как каждый раз время движения уменьшается вдвое на каждом шаге, то и круг положений каждый раз уменьшается в два раза, и поэтому через несколько повторений шага 2 получим круг положений с радиусом меньше r , что позволяет далее двигаться в центр круга положений с гарантированным уничтожением цели.

Литература

1. *Благодатских В.И.* Введение в оптимальное управление (линейная теория). М.: Высш. шк., 2001.
2. *Варга Дж.* Оптимальное управление дифференциальными и функциональными уравнениями. М.: Наука, 1977.
3. *Зубов В.И.* Лекции по теории управления. М.: Наука, 1975.
4. *Понтрягин Л.С., Болтянский В.Г., Гамкрелидзе Р.В., Мищенко Е.Ф.* Математическая теория оптимальных процессов. М.: Наука, 1961.
5. *Афанасьев В.Н., Колмановский В.Б., Носов В.Р.* Математическая теория конструирования систем управления. М.: Высш. шк., 2003.
6. *Алексеев В.М., Тихомиров В.М., Фомин С.В.* Оптимальное управление. М.: Наука, 1979.
7. *Гамкрелидзе Р.В.* Оптимальные по быстродействию процессы при ограниченных фазовых координатах // Изв. АН СССР. Сер. Матем. 1960. Т. 24. № 3. С. 315–356.
8. *Масина О.Н.* Вопросы управления движением транспортных систем // Транспорт: наука, техника, управление. 2006. № 12. С. 10–12.
9. *Масина О.Н.* Анализ оптимального управления транспортной системы // Устойчивость и колебания нелинейных систем управления: Тезисы докладов IX Международного семинара им. Е.С. Пятницкого. М.: Изд-во ИПУ РАН, 2006. С. 163–164.

Управление и стабилизация динамических систем на конечном интервале времени

Климачкова Т.С., аспирант, Мулкиджан А. С., аспирант

Московский государственный университет путей сообщения

Рассмотрена задача стабилизации системы до технической устойчивости, установлены теоремы об устойчивости на конечном интервале времени решения нелинейного уравнения при постоянно действующих возмущениях. Доказательства теорем базируются на использовании уравнения первого приближения, оценок В. М. Алексеева и логарифмической нормы матрицы.

Рассматривается нелинейное неавтономное дифференциальное уравнение

$$\dot{x} = g(t, x) \quad (1)$$

и соответствующее уравнению (1) возмущенное дифференциальное уравнение

$$\dot{y} = g(t, y) + \Phi(t, y). \quad (2)$$

предполагается, что вектор-функция $g(t, x)$ и ее производная $g'_x(t, x)$ непрерывны в некоторой области $G(t, x)$ пространства R^{n+1} , а возмущение $\Phi(t, x)$ удовлетворяет в области G условию Липшица относительно x .

Невозмущенное движение $x(t)=0$ системы (1) называется *технически устойчивым на заданном конечном интервале времени* $[t_0, T]$, если всякое возмущенное движение $x(t)$, определяемое начальным условием $x(t_0) \in S_0$, удовлетворяет условию $x(t) \in S(t) \quad \forall t \in [t_0, T]$, где $S_0(t) \subset R^n$ – множество начальных отклонений, $S(t)$ – множество допустимых значений отклонений $x(t)$ на $[t_0, T]$.

Невозмущенное движение $x(t)=0$ системы (2) называется *технически устойчивым на заданном конечном интервале времени* $[t_0, T]$ при наличии постоянно действующих возмущений $\Phi(t, x)$, если любое решение $x(t)$ системы (2), определенное начальным условием $x(t_0) \in S_0$, при любых постоянно действующих возмущениях $\Phi(t, x) \in W(t)$ удовлетворяет условию $x(t) \in S(t) \quad \forall t \in [t_0, T]$.

Вопросы устойчивости на конечном интервале времени и ее применение рассматривались в [1 – 3] и других работах.

Рассмотрим $(n \times n)$ -матрицу A и операторную норму $|A|$, индуцированную векторной нормой $|\cdot|$,

т.е. $|A| = \sup_{|x|=1} |Ax|$. Определим логарифмическую норму матрицы A

$$\mu(A) ::= \lim_{h \rightarrow 0^+} \frac{(|I + hA| - 1)}{h}, \quad (3)$$

где I – единичная $n \times n$ -матрица. Логарифмическая норма (3) в зависимости от выбора векторной и матричной нормы равна

$$\sup_i \left(\operatorname{Re} a_{ii} + \sum_{k \neq i} |a_{ik}| \right), \quad \sup_k \left(\operatorname{Re} a_{kk} + \sum_{i \neq k} |a_{ik}| \right) \quad \text{или} \quad \frac{1}{2} \Lambda(A + A^*),$$

где $\Lambda(C)$ – наибольшее собственное значение матрицы C .

Известно [4, 5], что если $x(t)$ – решение линейного дифференциального уравнения $\dot{x} = A(t)x$, то функции

$$|x(t)| \exp\left(-\int_0^t \mu(A) dt\right), \quad |x(t)| \exp\left(-\int_0^t \mu(-A) dt\right)$$

являются соответственно невозрастающими и неубывающими. Это свойство существенно используется для дальнейшего доказательства теорем об устойчивости на конечном интервале времени.

Пусть задано решение $\varphi_0^t x_0$ (решение, которое при $t = t_0$ обращается в x_0) уравнения (1) в некоторой заданной области, содержащей это решение, причем известно, что возмущение $\Phi(t, x)$ меньше некоторой величины $F(t)$. Если выполнены некоторые условия между заданными величинами и возмущениями начальных условий, при которых решение $\varphi_0^t y_0$ возмущенного уравнения (2) не покидает область D , то решение $\varphi_0^t x_0$ уравнения (1) является технически устойчивым (устойчивым на конечном интервале) при постоянно действующих возмущениях (в смысле Н.Д. Моисеева [2]).

Обозначим через $\frac{\partial x}{\partial y}$ матрицу $\begin{pmatrix} \frac{\partial x_i}{\partial y_j} \end{pmatrix}$, а $\varphi_\tau^t x$ – решение уравнения (1), для которого $\varphi_\tau^\tau \equiv x$. При сделанных выше предположениях функция $\varphi_\tau^t x$ дифференцируема по t, τ и x . По определению

$$\frac{\partial}{\partial t}(\varphi_\tau^t x) = g(t, \varphi_\tau^t x).$$

Обозначим

$$J(t, x) = \frac{\partial}{\partial x} g(t, x), \quad u(\tau, t, x) = \frac{\partial}{\partial \tau}(\varphi_\tau^t x), \quad A(x, \tau, t) = \frac{\partial}{\partial x}(\varphi_\tau^t x).$$

Нетрудно показать, что между u, J и A существует следующая связь

$$\frac{\partial A}{\partial t} = J(t, \varphi_\tau^t x)A, \quad A(x, \tau, \tau) = E, \quad \frac{\partial u}{\partial t} = J(t, \varphi_\tau^t x)u, \quad u(\tau, \tau, x) = -g(\tau, x).$$

Матрица A играет существенную роль в дальнейших рассуждениях. Если для некоторого $t > \tau$ и всех точек $\xi(s) = x_0 + (x - x_0)s$ справедливо неравенство

$$|A(\xi(s), \tau, t)| \leq H(s, \tau, t),$$

где $H(\cdot, \cdot, \cdot)$ – некоторая функция трех аргументов, то нетрудно понять, что справедлива и оценка

$$|\varphi_\tau^t x - \varphi_\tau^t y| \leq \int_0^1 H(s, \tau, t) |x - x_0| ds.$$

Семейство D_t множеств пространства R^n называется нормальным, если для каждого значения t и для каждой точки $x \in D_t$ существуют такая ее окрестность $N(x)$ и такое $t^* > t$, что $N(x) \subset D_{t^*} \quad \forall \tau \in (t, t^*)$.

Теорема 1. Решение $\varphi_0^t x$ дифференциального уравнения (1) технически устойчиво при постоянно действующих возмущениях, если для функций $\Phi(t, x)$, $A(x, t_1, t)$ и $\delta(x)$ выполнены условия:

- 1) $|\Phi(t, x)| \leq F(t)$ для всех $\forall (t, x) \in [t_0, T) \times D_t$;
- 2) для всех $x \in D_t$ и для всех t_1 и t таких, что для $t_0 < t_1 < t < T$ решение $\varphi_0^t x_0$ удовлетворяет условию $|A(x, t_1, t)| = \left| \frac{\partial}{\partial x} \varphi_0^t x \right| \leq R(t_1, t)$, где D_t – нормальное семейство на $[t_0, T)$;
- 3) $N(\varphi_0^t x_0, \delta(t)) \subset D_t \quad \forall t \in [t_0, T)$, где $N(x, \delta)$ – δ -окрестность точки x ;
- 4) $|y_0 - x_0| \leq \delta(t_0)$, где $\delta(t) = R(t_0, t)\delta(t_0) + \int_{t_0}^t R(s, t)F(s)ds$.

Показано, что при выполнении условий 1) – 4) теоремы 1 решение $\varphi_0^t y$ возмущенного дифференциального уравнения (2) определено для всех $t \in [t_0, T)$, причем

$$|\varphi_0^t y - \varphi_0^t x| \leq \delta(t),$$

т.е. решение $\varphi_0^t x$ уравнения (1) технически устойчиво при постоянно действующих возмущениях.

Теорема 2. Решение $\varphi_0^t x$ дифференциального уравнения (1) технически устойчиво при постоянно действующих возмущениях, если выполнены условия:

- 1) $q(t) \geq 0, F(t) \geq 0, \mu(t)$ – непрерывные функции, а $f(t)$ – дифференцируемая функция;
- 2) $|\Phi(t, y)| \leq F(t), y \in N_t$, где $N_t = N(\varphi_0^t x_0, f(t))$ – окрестность вокруг решения радиуса $f(t)$;
- 3) $\{(t, x) : t_0 \leq t \leq T, x \in N_t\} \subset G$, где $G = [t_0, t_1] \times D$;
- 4) $|J(t, y) - J(t, \varphi_0^t x_0)| \leq q(t)|y - \varphi_0^t x_0|, y \in N_t$;
- 5) $\mu(J(t, \varphi_0^t x_0)) \leq \mu(t)$;
- 6) $q'(t) \geq F(t) + \mu(t)f(t) + \frac{1}{2}q(t)f^2(t)$;
- 7) $|y_0 - x_0| \leq f(t_0)$.

Показано, что при выполнении условий 1) – 7) теоремы 2 на полуинтервале $[t_0, T)$ существует решение $\varphi_0^t y$ возмущенного уравнения (2), причем

$$|\varphi_0^t y - \varphi_0^t x| \leq z(t, |y_0 - x_0|) \leq f(t),$$

где $z(t, \lambda)$ – решение уравнения Риккати

$$\frac{\partial z}{\partial t} = F(t) + \mu(t)z + \frac{1}{2}q(t)z^2, z(t_0, \lambda) = \lambda.$$

Управление $u(t, x)$ называется стабилизирующим до технической устойчивости невозмущенное движение $x(t)=0$ системы

$$\dot{x} = f(t, x, u),$$

если при этом управлении $u(t, x)$ невозмущенное движение $x(t)=0$ системы технически устойчиво.

Такие управления называются стабилизирующими, а соответствующая задача – задачей стабилизации.

Задача стабилизации до технической устойчивости для конкретных технических систем (в частности, для посадочного аппарата «Сервейер» на участке торможения) рассматривалась в [8, 9]. В настоящей работе получены условия стабилизируемости до технической устойчивости моделей, описывающих движение железнодорожного экипажа, а также моделей движения тяжелой точки переменной массы.

Полученные результаты являются продолжением работ [6, 7] и представляют прикладной интерес для задач техники и естествознания, в частности, для задач оценки безопасности функционирования динамических систем.

Литература

1. Четаев Н.Г. Об одной мысли Пуанкаре // Сб. научн. тр. КАИ. 1935. № 3. С. 3 – 6.
2. Каменков Г.В. Об устойчивости движения на конечном интервале времени // ПММ. 1953. Т. 17. Вып. 5. С. 529 – 540.
3. Моисеев Н.Д. Обзор развития нелинейных теорий устойчивости // Труды ВВА им. Н.Е. Жуковского. Записки семинара по теории устойчивости движения. 1946. Вып. 1.
4. Алексеев В.М. Об одной оценке возмущений решений обыкновенных дифференциальных уравнений. I, II // Вестник Московского ун-та. 1961. № 2. С. 28 – 36; 1961. № 3. С. 3 – 10.
5. Лозинский С.М. Оценка погрешностей численного интегрирования обыкновенных дифференциальных уравнений // Изв. вузов. 1958. № 5. С. 52 – 90.
6. Климачкова Т.С. О технической устойчивости при постоянно действующих возмущениях // Applications of «Mathematica» system to social processes and mathematical physics. Proc. Of the International workshop. Брестский гос. ун-т, Беларусь; Wyzsza szkola finansow I zarzadzania w Sieldcach, Polska, 2003. P. 152 – 154.

-
7. *Климачкова Т.С.* Об алгоритмах исследования технической устойчивости математических моделей динамических систем // Тез. докладов XLIV Всерос. конф. по проблемам математики, информатики, физики и химии. М.: РУДН, 2008. С. 48.
 8. *Солодовников В.В., Семенов В.В.* Спектральная теория нестационарных систем управления. М.: Наука, 1974.
 9. *Тарасов А.П.* Алгоритмы, основанные на методе функций Ляпунова и теории разностных схем, обеспечивающие техническую устойчивость динамических систем. Дисс. ... канд. ф.-м. наук. М.: МАИ (ГТУ), 1998.

О природе турбулентности в конвекции Рэлея – Бенара

Магницкий Н. А., д.ф.-м.н., проф., акад. РАН

*Институт системного анализа РАН, Москва
117312, г. Москва, пр-т 60-летия Октября, 9, E-mail: nmag@isa.ru*

Аннотация

Проблема турбулентности названа институтом математики Клэя одной из семи математических проблем тысячелетия и находится также в списке сформулированных С.Смейлом 18 наиболее значимых математических проблем XXI века. Огромные усилия, уделяемые во всем мире в последние десятилетия фундаментальным и прикладным исследованиям проблемы турбулентности, показали, что ее решение невозможно исключительно методами чистой математики без привлечения быстродействующей электронно-вычислительной техники, разработке и созданию которой была посвящена научная деятельность академика В.А.Мельникова. В представленной работе предложен оригинальный метод теоретического и численного анализа движения вязкой несжимаемой жидкости при переходе к турбулентному режиму в конвекции Рэлея – Бенара. Показано, что на начальной стадии развитие турбулентности в фазовом пространстве задачи происходит через субгармонический каскад бифуркаций устойчивых двумерных торов в соответствии с универсальным сценарием Фейгенбаума–Шарковского–Магницкого.

Решение

Международной научной конференции «Проблемы вычислительной техники и программирования», посвященной 80-летию со дня рождения академика В.А.Мельникова

Организаторы Конференции: Российская академия наук – Институт системного программирования, НИИ системных исследований, Межведомственный суперкомпьютерный центр, Научный фонд «Первая исследовательская лаборатория имени академика В. А. Мельникова».

Участники Конференции: академики и члены-корреспонденты РАН, ведущие ученые ВУЗов и научно-производственных предприятий России, ученые и исследователи Беларуси, Украины, Грузии, Болгарии, Германии, Китая, Вьетнама.

29 января 2009 года была проведена Научная сессия Отделения нанотехнологии и информационных технологий РАН, посвященная 80-летию со дня рождения академика В. А. Мельникова. Видные ученые представили 8 научных докладов на актуальные темы по вычислительным наукам.

19-20 февраля 2009 года проходила международная научная конференция, посвященная 80-летию со дня рождения академика В. А. Мельникова, которая транслировалась в Интернет в онлайн режиме. В рамках Конференции проведены пленарное и секционные заседания, на которых заслушаны 56 научных докладов, посвященных перспективным направлениям развития вычислительной техники и программирования. Присутствовало около 200 участников. Группа ученых была награждена медалью «За высокие достижения в науке» Научного фонда «Первая исследовательская лаборатория имени академика В. А. Мельникова»

Конференция получила поздравительные послания от Государственной думы РФ, Бюро ЦК КПРФ, Министра образования и науки РФ, Мэра города Москвы, Политехнического музея РФ, Института вычислительной технологии Китайской академии наук.

20 февраля в 12:34 в прямом эфире по громкой связи с трансляцией в зал заседания научной секции «Вычислительные науки и проблемы системной безопасности» Конференцию приветствовал Командир международной космической станции герой России Ю. В. Лончаков от имени интернационального экипажа, действующего на космической орбите.

Конференция решила:

1. Издать сборник научных докладов по материалам Конференции, а также книгу «Вычислительные науки и проблемы системной безопасности» по материалам секции.
2. Утвердить новый состав Попечительского совета некоммерческой организации Научного фонда «Первая исследовательская лаборатория имени академика В. А. Мельникова»:
ЖИЖЧЕНКО А. Б. - чл.-корр. РАН, заместитель академика-секретаря ОМН РАН;
САВИН Г. И. - академик, директор Межведомственного суперкомпьютерного центра РАН;
ПЛАТОНОВ В. П. - академик, зав. отделом НИИСИ РАН;
ВИНОГРАДОВ В. Ю. - заместитель мэра г. Москвы;
МЕЛЬНИКОВ И. И. - профессор, депутат Госдумы РФ;
БУГАЕВ А. С. - академик, зав. кафедрой МФТИ, зав. Лабораторией ИРЭ РАН.
3. Подготовить видеоматериалы Конференции и видеокадры деятельности академика В. А. Мельникова на DVD-носителях, организовать передачу на ТВ-канале «Культура».
4. Просить Президиум РАН рассмотреть вопрос об учреждении Премии имени академика В. А. Мельникова РАН за выдающиеся работы в области вычислительной техники и программирования.
5. Увековечить имя академика В. А. Мельникова путем установления мемориальных досок на доме, где жил В. А. Мельников, на парадный вход Института, где работал В. А. Мельников, и на родине В. А. Мельникова в Чеховский район Московской области.
6. Открыть главу, посвященную академику В. А. Мельникову, в электронной библиотеке РАН «Научное наследие России».

Оргкомитет Конференции благодарит всех, принимавших участие в работе Международной научной конференции, посвященной 80-летию академика В. А. Мельникова.

Сопредседатели пленарного заседания Конференции

А. Н. Томилин, М. Ш. Сургуладзе

г. Москва, 19-20 февраля 2009 года.



МЭР МОСКВЫ
ЮРИЙ МИХАЙЛОВИЧ ЛУЖКОВ

19 февраля 2009 г.

Организаторам, участникам и гостям Международной научной конференции, посвящённой 80-летию со дня рождения академика В.А. Мельникова

Дорогие друзья!

От имени Правительства Москвы и от себя лично горячо приветствую всех организаторов, участников и гостей Международной научной конференции, посвящённой 80-летию со дня рождения Владимира Андреевича Мельникова, выдающегося учёного и конструктора, мудрого воспитателя целого ряда поколений разработчиков отечественных электронных вычислительных машин, систем и комплексов!

Его судьба – это не только вклад в развитие отечественной науки, выразившийся в создании высокоскоростной ЭВМ и руководстве организованным им в Москве Институтом проблем кибернетики, но также пример самоотверженного служения любимому делу и Родине, имеющий необычайно важное значение для сегодняшней России.

Убеждён, что Конференция внесёт весомый вклад в дело, которому посвятил свою жизнь академик В.А. Мельников, и станет заметным этапом в истории российской кибернетики.

Желаю всем активной, плодотворной работы, полезных деловых контактов и новых успехов на благо отечественной науки и процветания Москвы и России!



ф. ТГ-17



ПРАВИТЕЛЬСТВЕННАЯ ТЕЛЕГРАММА

Присл: <u>18/2</u> <u>1932</u>	Для заметок адресата
Бланк №	
Понял:	



ТЕЛЕГРАММА

МОСКВА 265/3/13501 108 18/2 1930=

ПРАВИТЕЛЬСТВЕННАЯ Г МОСКВА УЛ КРАСНОПРУДНАЯ 36-1-13 ПРЕЗИДЕНТУ
 НАУЧНОГО ФОНДА "ПЕРВАЯ ИССЛЕДОВАТЕЛЬСКАЯ ЛАБОРАТОРИЯ ИМ АКАДЕМИКА
 В А МЕЛЬНИКОВА" СУРГАЛАДЗЕ М Ш=

СЕРДЕЧНО ПРИВЕТСТВУЮ ОРГАНИЗАТОРОВ УЧАСТНИКОВ И ГОСТЕЙ
 МЕЖДУНАРОДНОЙ НАУЧНОЙ КОНФЕРЕНЦИИ ПОСВЯЩЕННОЙ 80-ЛЕТИЮ СО ДНЯ
 РОЖДЕНИЯ ВЫДАЮЩЕГОСЯ УЧЕНОГО И КОНСТРУКТОРА ВЛАДИМИРА АНДРЕЕВИЧА
 МЕЛЬНИКОВА

ПОЛАГАЮ ЧТО КОНФЕРЕНЦИЯ БУДЕТ СПОСОБСТВОВАТЬ ДАЛЬНЕЙШЕМУ ПРОЦЕССУ
 РАЗВИТИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ВНЕСЕТ ВЕСОМЫЙ ВКЛАД В ДЕЛО
 КОТОРОМУ ПОСВЯТИЛ СВОЮ ЖИЗНЬ ВЛАДИМИР АНДРЕЕВИЧ

ЖЕЛАЮ ВСЕМ УЧАСТНИКАМ ДАННОГО МЕРОПРИЯТИЯ ДАЛЬНЕЙШИХ УСПЕХОВ И ЧТО
 ОСОБЕННО ВАЖНО УДАЧНОГО ВНЕДРЕНИЯ ВАШИХ РЕШЕНИЙ И ИДЕЙ В ИНТЕРЕСАХ
 ОБЩЕГО ДЕЛА=ПРЕДСЕДАТЕЛЬ КОМИТЕТА ГД ФС РФ ПО НАУКЕ И НАУКОЕМКИМ
 ТЕХНОЛОГИЯМ В А ЧЕРЕШНЕВ

ИСХ НР 3.30-36/45 18 ФЕВРАЛЯ 2009 Г-

НННН 1932 18.02 0051

МИНИСТЕРСТВО КУЛЬТУРЫ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО КУЛЬТУРЕ И КИНЕМАТОГРАФИИ



Федеральное государственное учреждение культуры

101000, Москва, Центр,
Новая пл., д. 3/4, подъезд 1.
Http://www.polymus.ru
Тел.: +7(495)625-0614;
+7(495)625-0107;
Факс: +7(495)625-1290

ПОЛИТЕХНИЧЕСКИЙ МУЗЕЙ

18.02.09 № 09-10/122

на № _____

Уважаемые участники
международной научной конференции
«Проблемы вычислительной техники и программирования»,
посвященной 80-летию
академика Владимира Андреевича Мельникова!

Политехнический музей высоко ценит заслуги академика В.А. Мельникова и его вклад в развитие отечественной электронной вычислительной техники.

Владимир Андреевич Мельников является одним из тех, кому наша страна обязана своим вхождением в мир ЭВМ, который открылся человечеству во всей своей полноте во второй половине XX века. Политехнический является, пожалуй, единственным государственным музеем, который формирует и использует в интересах воспитания и образования культурное наследие народов нашей страны в области техники и научно-технических знаний.

Наш музей располагает богатой коллекцией средств вычислительной техники, имеющей, несомненно, историческое значение. Кроме того, сформирован богатый документальный фонд по этой теме. Нам приятно сообщить, что в собрании музея имеются документальные материалы о научной деятельности академика В.А. Мельникова.

В своей дальнейшей работе по выявлению и вовлечению в музейный фонд исторических объектов и документов, связанных с шагами нашей страны в мире вычислительной техники, мы могли бы опираться на Вашу поддержку.

С уважением,
Генеральный директор



Г.Г. Григорян



Участникам
международной конференции

Уважаемые коллеги!

Приветствую участников международной конференции «Проблемы вычислительной техники и программирования», посвященной 80-летию выдающегося ученого конструктора Владимира Андреевича Мельникова.

Отечественные вычислительные системы, созданные научной школой академика В.А. Мельникова, достойно, на высоком научно-техническом уровне обеспечивают решение чрезвычайно важных научных, оборонных и народнохозяйственных задач, являются образцом плодотворного сотрудничества фундаментальной науки, высокотехнологичного сектора промышленности и образования.

Российские супер-ЭВМ, под получившим признание ученых всего мира брендом БЭСМ, имеют уникальную архитектуру, математическое обеспечение и, что особенно ценно, созданы на отечественной элементной базе.

Убежден, что работа форума будет исключительно конструктивной, а широкая и открытая научная дискуссия даст новый импульс исследовательской работе, устремлениям и достижениям в ключевых областях знаний, обогатит интеллектуальный и творческий потенциал участников.

Желаю всем участникам удивительных открытий и новых достижений!

Министр

A handwritten signature in black ink, consisting of a stylized, cursive 'A' followed by a horizontal line and a small flourish.

А.А. Фурсенко



ПРАВИТЕЛЬСТВЕННАЯ ТЕЛЕГРАММА

Присл: 10/11 го 18 час 18 мин.	Для заметок адресата
Бланк № 00401	
Принял: [подпись] 27	

ТЕЛЕГРАММА



МОСКВА 265/4/5201 123 10/2 1815=

ПРАВИТЕЛЬСТВЕННАЯ МОСКВА УЛ ВАВИЛОВА ЧО ВЦ РАН ОРГКОМИТЕТ
МЕЖДУНАРОДНОЙ КОНФЕРЕНЦИИ ВИЦЕ-АДМИРАЛУ ПРОФЕССОРУ Н А СЕВЕРЦЕВУ=

УВАЖАЕМЫЕ ТОВАРИЩИ

ПРИВЕТСТВУЮ ОРГКОМИТЕТ И ВСЕХ УЧАСТНИКОВ КОНФЕРЕНЦИИ ПОСВЯЩЕННОЙ
80-ЛЕТИЮ ВЫДАЮЩЕГОСЯ УЧЕНОГО И КОНСТРУКТОРА НАСТОЯЩЕГО
ПАТРИОТА-ВЛАДИМИРА АНРЕЕВИЧА МЕЛЬНИКОВА. СОЗДАННЫЕ ИМ
ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ ОПЕРЕЖАЛИ СВОЕ ВРЕМЯ И ПРЕВОСХОДИЛИ
ЗАРУБЕЖНЫЕ АНАЛОГИ. ЕГО НАУЧНАЯ ШКОЛА В ОБЛАСТИ ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКИ И ПРОГРАММИРОВАНИЯ ДАЛА НАШЕЙ РОДИНЕ КВАЛИФИЦИРОВАННЫХ
СПЕЦИАЛИСТОВ УЧЕНИКОВ И ПОСЛЕДОВАТЕЛЕЙ КОТОРЫЕ ДАЖЕ В НЫНЕШНЕЕ
ТРУДНОЕ ВРЕМЯ РАЗВИВАЮТ НАСЛЕДИЕ ОСТАВЛЕННОЕ В А МЕЛЬНИКОВЫМ
ВЫРАЖАЮ ГЛУБОЧАЙШУЮ ПРИЗНАТЕЛЬНОСТЬ И БЛАГОДАРНОСТЬ ВАМ ЕГО
УЧЕНИКАМ ДРУЗЬЯМ И ВСЕМ ПАТРИОТАМ НАШЕЙ РОДИНЫ ЗА СОХРАНЕНИЕ
ПАМЯТИ И НАУЧНОГО НАСЛЕДИЯ ВЕЛИКОГО УЧЕНОГО И ЖЕЛАЮ УСПЕХОВ В
ПРОВЕДЕНИИ КОНФЕРЕНЦИИ С УВАЖЕНИЕМ=РУКОВОДИТЕЛЬ ФРАКЦИИ КПРФ В
ГОСУДАРСТВЕННОЙ ДУМЕ РФ ПРЕДСЕДАТЕЛЬ ЦК КПРФ Г А ЗЮГАНОВ

10 ФЕВРАЛЯ 2009 Г-

НННН 1818 10.02 0027



ГОСУДАРСТВЕННАЯ ДУМА
ФЕДЕРАЛЬНОГО СОБРАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЯТОГО СОЗЫВА

ДЕПУТАТ
ГОСУДАРСТВЕННОЙ ДУМЫ
(2008–2011)

18 02 20 09 г.

№ ДДБ-25/09

Участникам Международной
научной конференции,
посвящённой 80-летию
со дня рождения академика
В.А. Мельникова

От всей души приветствую участников и организаторов Конференции, посвященной 80-летию выдающегося ученого и конструктора академика Мельникова Владимира Андреевича.

Великая заслуга академика Мельникова В.А. и научного коллектива, руководимого им, состоит в разработке и создании отечественных быстродействующих электронных машин, комплексов и систем, которые явились основным средством обработки информации, используемых в научных исследованиях, промышленности и обороне нашей страны. Ими были заложены прочные традиции объединения ученых, инженеров и конструкторов в интересах развития отечественного производства.

Сегодня, для перехода России на инновационный путь развития, перед нашей отечественной наукой поставлена задача по разработке наукоемких технологий и внедрению их в промышленное производство.

При этом первоочередной задачей является образование и воспитание молодежи, приобщение её к профессиональной работе на благо нашей страны и общества на пути к устойчивому развитию.

Председатель Подкомитета
по проблемам устойчивого развития России ГД ФС РФ,
Герой Социалистического Труда,
Академик РАН

М.Ч. Залиханов