

Федеральное государственное учреждение «Федеральный научный центр  
Научно-исследовательский институт системных исследований Российской академии наук»  
(ФГУ ФНЦ НИИСИ РАН)

## **ТРУДЫ НИИСИ РАН**

ТОМ 8 № 5

**МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ  
СЛОЖНЫХ СИСТЕМ:**

**ТЕОРЕТИЧЕСКИЕ И ПРИКЛАДНЫЕ АСПЕКТЫ**

МОСКВА  
2018

**Редакционный совет ФГУ ФНЦ НИИСИ РАН:**

В.Б. Бетелин (председатель),  
Е.П. Велихов, В.А. Галатенко, В.Б. Демидович (отв. секретарь), Б.В. Крыжановский,  
А.Г. Кушниренко, А.Г. Мадера, М.В. Михайлюк, В.Я. Панченко, В.П. Платонов,  
В.Н. Решетников

**Главный редактор журнала:**

В.Б. Бетелин

**Научный редактор номера:**

А.Г.Мадера

**Тематика номера:**

Моделирование и проектирование электронных устройств, информационные и компьютерные технологии, тепловые процессы в электронных системах, физические процессы и их моделирование, математические исследования.

Журнал публикует оригинальные статьи по следующим областям исследований: математическое и компьютерное моделирование, обработка изображений, визуализация, системный анализ, методы обработки сигналов, информационная безопасность, информационные технологии, высокопроизводительные вычисления, оптико-нейронные технологии, микро- и нанoeлектроника, математические исследования и вопросы численного анализа, история науки и техники.

**The topic of the issue:**

Modeling and design of electronic devices, information and computer technology, thermal processes in electronic systems, physical processes and their modeling, mathematical studies.

The Journal publishes novel articles on the following research areas: mathematical and computer modeling, image processing, visualization, system analysis, signal processing, information security, information technologies, high-performance computing, optical-neural technologies, micro- and nanoelectronics, mathematical researches and problems of numerical analysis, history of science and of technique.

Заведующий редакцией: Ю.Н.Штейников

Издатель: ФГУ ФНЦ НИИСИ РАН,  
117218, Москва, Нахимовский проспект 36, к. 1

## СОДЕРЖАНИЕ

### I. МОДЕЛИРОВАНИЕ И ПРОЕКТИРОВАНИЕ ЭЛЕКТРОННЫХ УСТРОЙСТВ

<i>А.С.Бакулин, С.А.Бахов, О.В.Сердин, А.Г.Прозорова, И.В.Токарев.</i> Методы ограничения ЭМП электронных модулей и ЭВМ типа сервер .....	5
<i>А.А.Подковыров.</i> Методика разработки универсального керамического корпуса микросхемы .....	11
<i>А.А.Захарченко, Е.П.Михальцов, М.В.Орешков, В.Ю.Троицкий.</i> Повышение чувствительности установки контроля дефектов в светлом поле при наличии вариаций яркости изображений .....	19

### II. ИНФОРМАЦИОННЫЕ И КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ

<i>А.Б.Бетелин, А.А.Прилипка, А.Г.Прилипка, Г.А.Прилипка, Д.В.Самборский.</i> Методы построения систем высокой готовности .....	25
<i>А.Б.Бетелин, А.А.Прилипка, Г.А.Прилипка, С.Г.Романюк, Д.В.Самборский.</i> Оптимизация архивирования виртуальных операционных систем в среде CNU Linux/QEMU/KVM/Libvirt .....	32
<i>А.Н.Годунов, В.А.Солдатов.</i> Методы и средства оценки объёма памяти, требуемой для выполнения образа ОСРВ Багет .....	42
<i>А.Н.Сотников, И.Н.Соболевская, И.Н.Чередниченко.</i> О некоторых подходах к построению цифровых моделей 3D-объектов и формированию виртуальных коллекций .....	53
<i>А.В.Крыницкий.</i> Аппаратная реализация алгоритма Adler 32 для вычисления контрольных сумм .....	58
<i>А.В.Мальцев, Д.В.Омельченко.</i> Модель расчёта отражённого света для имитации на GPU физически корректного освещения объектов виртуальной среды .....	65
<i>А.В.Мальцев, П.Ю.Тимохин, Д.В.Омельченко.</i> Технология моделирования и управления источниками искусственного освещения МКС в системах виртуального окружения .....	70
<i>Е.В.Страшинов, Л.А.Финагин.</i> Моделирование динамики манипуляционных роботов в системах виртуального окружения .....	76
<i>П.Ю.Тимохин, М.В.Михайлюк.</i> Эффективные алгоритмы визуализации результатов виртуального моделирования ядра с использованием шейдерной тесселяции .....	83

### III. ТЕПЛОВЫЕ ПРОЦЕССЫ В ЭЛЕКТРОННЫХ СИСТЕМАХ

<i>А.В.Бакалдин, П.И.Кандалов, Н.А.Козлов, А.Г.Мадера, О.В.Сердин.</i> Методы обеспечения отвода тепла в корпусах высокопроизводительных защищённых серверов с воздушным охлаждением .....	90
<i>А.В.Бакалдин, А.М.Баранов, П.И.Кандалов, А.Г.Мадера, О.В.Сердин.</i> Современные технологии охлаждения с применением контурных тепловых труб в корпусах бортовых серверов с повышенной мощностью потребления .....	98
<i>М.Ж.Акжолов, П.И.Кандалов, А.Г.Мадера.</i> Исследование влияния параметров теплового режима на статистические меры стохастической температуры электронного модуля .....	106
<i>А.Г.Мадера.</i> Тепловой режим микросхемы в условиях неопределённости определяющих параметров и воздействии тепловой обратной связи .....	110

<i>А.Г.Мадера.</i> Метод стохастического псевдорешения при математическом моделировании тепловых режимов в электронных системах в условиях неопределённости определяющих параметров .....	115
<i>А.А.Воробьёв, П.И.Кандалов, А.Г.Мадера, О.В.Сердин.</i> Принципы охлаждения корпусов моноблоков повышенной мощности потребления с применением тепловых труб .....	121
<i>А.В.Зуйков, А.А.Воробьёв.</i> Исследование влияния параметров клинового механизма на обеспечение теплоотвода модуля 6U в магистрально-модульной системе типа VPX .....	125

#### **IV. ФИЗИЧЕСКИЕ ПРОЦЕССЫ И ИХ МОДЕЛИРОВАНИЕ**

<i>С.Г.Вольпин, А.Г.Дяченко, Н.П.Ефимова, О.В.Ломакина, Ю.Б.Чен-лен-сон.</i> Обзор новых интерпретационных моделей при гидродинамических исследованиях скважин .....	131
<i>Н.А.Агафонов, Н.В.Масальский, В.В.Осипов, А.В.Родителев.</i> Проблемы моделирования переноса в квазипланарных КНИ КМОП нанотранзисторах .....	139
<i>В.Г.Редько.</i> Моделирование процессов эволюции популяций конкурирующих автономных агентов .....	148

#### **V. МАТЕМАТИЧЕСКИЕ ИССЛЕДОВАНИЯ**

<i>А.С.Осипов.</i> О нечётких алгоритмах ближайших соседей .....	153
<i>Д.В.Гулуа, Е.К.Гребенникова.</i> Оценка погрешности приближённого решения с применением ассоциированных полиномов .....	162
<i>Мзия Цхомелидзе.</i> Аффинные преобразования и координатизация (аксиоматика) для модулей над общими кольцами .....	165

# Методы ограничения ЭМП электронных модулей и ЭВМ типа сервер

А.С. Бакулин<sup>1</sup>, С.А. Бахов<sup>2</sup>, О.В.Сердин<sup>3</sup>, А.Г. Прозорова<sup>4</sup>, И.В. Токарев<sup>5</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

E-mail's: <sup>1</sup>[bakulin.as@niisi.msk.ru](mailto:bakulin.as@niisi.msk.ru), <sup>2</sup>[bakhov@cs.niisi.ras.ru](mailto:bakhov@cs.niisi.ras.ru), <sup>3</sup>[serdin@cs.niisi.ras.ru](mailto:serdin@cs.niisi.ras.ru),

<sup>4</sup>[prozorova@cs.niisi.ras.ru](mailto:prozorova@cs.niisi.ras.ru), <sup>5</sup>[tokigvl@cs.niisi.ras.ru](mailto:tokigvl@cs.niisi.ras.ru)

**Аннотация:** В данной работе представлен анализ основных применяемых методов ограничения электромагнитных помех. Предложены методики и общие принципы конструирования электронных модулей для защищенных серверов, а также методики уменьшения помех в корпусах ЭВМ в стоечном исполнении. Эффективность предложенных методик подтверждена экспериментальными данными, полученными на альтернативной измерительной площадке ФГУ ФНЦ НИИСИ РАН.

**Ключевые слова:** корпус защищенного сервера, принципы ограничения электромагнитных помех, отечественный двухядерный 64-х разрядный микропроцессор, воздушное охлаждение.

## 1. Введение

Современный уровень и тенденции развития электронных устройств предполагают высокие скорости обработки цифровых сигналов, у которых длина волн гармоник меньше или соизмерима с размерами печатных плат (ПП) и длинами сигнальных трасс. Для ограничения ЭМП требуется учет эффектов распространения электромагнитных волн, целостности сигналов, задержки распространения сигналов, и повышенное внимание к вопросам электромагнитной совместимости (ЭМС), помехозащищенности и вопросам согласования линий передач сигналов.

При разработке, проектировании и модификации электронных модулей и ЭВМ типа сервер должны учитываться требования в области ЭМС и защиты информации, которые могут быть предъявлены к техническим средствам (ТС), содержащим разрабатываемые модули и ЭВМ. Широкая область применения и универсальность разрабатываемых модулей и ЭВМ типа сервер предполагает возможность предъявления к ТС самых жестких требований по электромагнитной совместимости (ЭМС) и специальных требований по защите информации (СТР).

## 2. Нормы обеспечения функциональной безопасности технических средств

Методологию обеспечения функциональной безопасности технических средств в

отношении промышленных радиопомех определяет ГОСТ В25803-91. Допустимые уровни напряжения радиопомех на отходящих от ТС проводах и кабелях (кондуктивные помехи), и допустимые уровни напряженности полей, которые излучаются ТС и отходящими от него проводами и кабелями, определяются условиями эксплуатации изделия. В частности, наиболее жесткие требования по уровням излучений промышленных помех предъявляются к устройствам электронно-вычислительной техники, работающим в подвижных и стационарных пультах связи совместно с радиоприемными устройствами.

Методологию обеспечения функциональной безопасности технических средств в отношении внешних электромагнитных помех определяет ГОСТ Р 52317.1.2-2007 (МЭК 61000-1-2:2001).

При отсутствии предъявленных конкретно к ТС требований и стандартов ЭМС, к нему применяются общие стандарты ЭМС, достаточные для обеспечения приемлемого уровня помехоустойчивости, которые можно рассматривать в качестве обобщенных характеристик электромагнитной обстановки.

Специальные требования по защите информации от утечки по техническим каналам определяются ГОСТ Р 29339-92, ГОСТ Р 50752-95, ГОСТ Р 53115-2008 и руководящими документами ФСТЭК России.

Основные требования к корпусам изделий, к которым предъявляются требования по защите информации определяются в ГОСТ Р 50543-93.

### 3. Основные применяемые методы ограничения помех

В соответствии с ГОСТ Р 51317.1.2-2007 (“Методология обеспечения функциональной безопасности электрического и электронного оборудования в отношении электромагнитных помех”) применяются технические средства предотвращения проникновения и распространения электромагнитных помех в следующей последовательности:

- а) экранирование;
- б) фильтрация;
- в) защита от перенапряжений;
- г) заземление и эквипотенциальность соединений;
- д) обеспечение помехоустойчивости при проектировании печатных плат.

#### 3.1 Экранирование

При конструировании корпусов и шкафов в стоечном исполнении ЭВМ рекомендуется учитывать требования ГОСТ Р 50543-93 для корпусов по классу 1. В частности, принять за основу значения эффективности экранирования корпусов в соответствии с приведенной ниже таблицей 1.

Таблица 1. Рекомендуемая эффективность экранирования корпусов и шкафов

Диапазон частот, МГц	Эффективность экранирования, дБ, не менее
До 20	30
20—300	60
300—1000	90
1000—10000	90...100

Для обеспечения вышеприведенных требований экранирования применяется электропроводящее покрытие всех элементов разрабатываемых корпусов и стоек, а также применяется сопряжение всех элементов корпуса между собой по поверхностям сопряжения.

Суммарная эффективность экранирования корпусов определяется расчетным путем для потенциально самых слабых мест конструкции корпуса (крышек, дверок, соединителей).

Размеры сопрягающих поверхностей элементов корпусов и шкафов в стоечном исполнении ЭВМ, а также элементов крепления, рассчитываются с учетом техники заградительных волноводов для самых высоких частот гармоник помех.

Монтаж съемных крышек и дверок осуществляется с использованием специальных экранирующих уплотнителей.

Выбор соединителей для интерфейсных сигналов учитывает возможность их монтажа на корпус с минимальным переходным импедансом во всем диапазоне частот, а также возможность экранирования сигналов в отходящих кабелях.

Выбор соединителей высокочастотных интерфейсов учитывает согласование сигнальных трасс по волновому сопротивлению с минимумом потерь.

Для неиспользуемых интерфейсных соединителей предусмотрена установка экранирующих заглушек.

#### 3.2 Фильтрация

При разработке модуля питания обеспечивается минимально возможный уровень несимметричных помех, путем оптимизации разводки и фильтрации помех на корпусах преобразователей напряжения и их радиаторов.

На входе модуля питания устанавливаются первичные входные фильтры симметричных и несимметричных радиопомех.

Допустимые уровни несимметричных промышленных радиопомех всей системы питания ТС достигаются установкой экранированной сетевой развязки в районе ввода (соединителя) питания внутри корпуса или шкафа в стоечном исполнении ЭВМ. Сетевая развязка обеспечивает гарантированное подавление несимметричных промышленных радиопомех в сети питания на частотах более 10 МГц.

При необходимости, допустимый уровень симметричных промышленных радиопомех может достигаться за счет установки дополнительного фильтра, установленного внутри или снаружи корпуса или шкафа. Уровень затухания и частотный диапазон фильтра должен рассчитываться после измерения реальных промышленных радиопомех ТС и сравнения их с уровнями допустимых промышленных радиопомех.

При использовании низкоскоростных интерфейсов допускается возможность установки при необходимости соединителей с фильтрами контактами.

Уровень электромагнитных помех в цепях питания на печатных платах контро-

лируется установкой высокочастотных блокирующих конденсаторов с применением методов моделирования.

При необходимости уменьшения уровней излучения промышленных радиопомех за счет отходящих кабелей необходимо предусматривать установку на их экраны кабельных ферритовых фильтров.

### 3.3 Защита от перенапряжений

Схемотехника сетевой развязки подразумевает установку предохранителя и электронных быстродействующих элементов защиты от перенапряжения.

Защита от внешних низкочастотных перенапряжений на экранирующих оплетках отходящих кабелей осуществляется за счет гальванической развязки сигнальной земли, которая обеспечивает превентивную защиту ТС от внешних электромагнитных воздействий большой мощности в области низких частот до 10...30 МГц (например, молния, ядерный взрыв) и защиту технического персонала от поражения током

### 3.4 Заземление и эквипотенциальность соединений

При конструировании корпусов ТС предусматриваются специальные места заземления, обеспечивающие надежное некорродирующее соединение с внешней системой заземления.

При конструировании корпусов используется минимальное количество сопрягаемых деталей и обеспечивается минимальная индуктивность путей возвратных токов от соединителей до источников помех на печатных платах.

Конструкция стоек для ЭВМ в стоечном исполнении предусматривает многоточечное соединение элементов стоек по плоскостям прилегания с внутренней поверхностью шкафа.

По контуру печатных плат модулей предусмотрены полигоны корпусной земли, которые с минимальным переходным импедансом соединяются через корпусные элементы модулей с внутренней поверхностью корпусов ТС. Указанные полигоны соединяются на печатных платах с плоскостями системной (сигнальной) земли через высокочастотные конденсаторы, расположенные по периметру с учетом максимальных частот гармоник сигналов, присутствующих в модуле. Это обеспечивает эквипотенциальность системной земли и внутренней поверхности

корпусов ТС во всем диапазоне частот, а, следовательно, уменьшение уровней напряжения несимметричных помех в бюджете шумов на печатных платах.

В объединительных платах предусматриваются наличие по контуру полигонов корпусной земли. Полигоны корпусной земли соединяются с контактными площадками, предусмотренными в корпусе ТС.

### 3.5 Обеспечение помехоустойчивости при проектировании печатных плат

При разработке электронных модулей применяются методы подавления электромагнитных помех в диапазоне частот от постоянного тока до 10 ГГц.

Выбор стандарта модулей VPX обеспечивает межмодульную согласованную передачу сигналов (в том числе дифференциальных) во всем требуемом диапазоне частот с минимальными потерями.

Разработка печатных плат модулей и объединительных плат ведется с применением САПР фирмы CADENCE, позволяющей разрабатывать многослойные и высокоскоростные печатные платы с высокой плотностью размещения компонентов. Анализ целостности сигналов и ЭМС топологии печатных плат выполняется с помощью специального модуля SPECCTRAQuest SI Expert с использованием SPACE и IBIS моделей компонентов. Дополнительно проверяется импеданс, эквипотенциальность и отсутствие резонансов в полигонах питания и земли во всем диапазоне частот.

При выборе стека слоев задается контроль разводки сигнальных трасс по волновому сопротивлению и учитывается необходимость не только непрерывности опорных плоскостей, но и необходимость экранирования ими внутренних и интерфейсных сигналов.

Отдается предпочтение дифференциальной передаче сигналов, обеспечивающей минимальный уровень электромагнитных помех.

## 4. Результаты измерений

Проводились измерения напряжения промышленных радиопомех опытного образца ЭВМ. Испытания проводились на альтернативной измерительной площадке ФГУ ФНЦ НИИСИ РАН. Результаты измерения приведены ниже в таблице 2 и на рисунке 1.

Таблица 2. Напряжения промышленных радиопомех опытного образца ЭВМ

$F_i$ МГц	$U_{изм}$ дБ мкВ	$K$ дБ	$U_{ном}$ дБ мкВ	$U_{ном}$ дБ мкВ	Превышение дБ мкВ
0.10	40.20	0.00	40.20	61.16	-
0.14	40.20	0.00	40.20	56.55	-
0.21	82.20	0.00	82.20	72.78	9.42
0.28	82.70	0.00	82.70	72.00	10.70
0.33	76.20	0.00	76.20	71.06	5.14
0.41	66.20	0.00	66.20	68.82	-
0.55	57.20	0.00	57.20	65.98	-
0.61	58.20	0.00	58.20	66.28	-
0.67	66.20	0.00	66.20	66.51	-
0.82	62.20	0.00	62.20	67.19	-
0.95	41.20	0.00	41.20	67.80	-
1.23	52.20	0.00	52.20	65.06	-
1.37	49.20	0.00	49.20	65.43	-
3.89	52.20	0.00	52.20	61.64	-
5.71	41.20	0.00	41.20	60.12	-
8.45	44.20	0.00	44.20	60.00	-
10.60	39.70	0.00	39.70	60.00	-
16.80	34.70	0.00	34.70	60.00	-
19.10	30.90	0.00	30.90	60.00	-
23.00	37.70	0.00	37.70	60.00	-
25.50	23.70	0.00	23.70	60.00	-
39.00	21.20	0.00	21.20	68.00	-
40.00	17.20	0.00	17.20	68.00	-
50.00	17.20	0.00	17.20	68.00	-
61.00	15.20	0.00	15.20	68.00	-
71.00	12.20	0.00	12.20	68.00	-
82.00	9.60	0.00	9.60	68.00	-
92.00	9.70	0.00	9.70	68.00	-

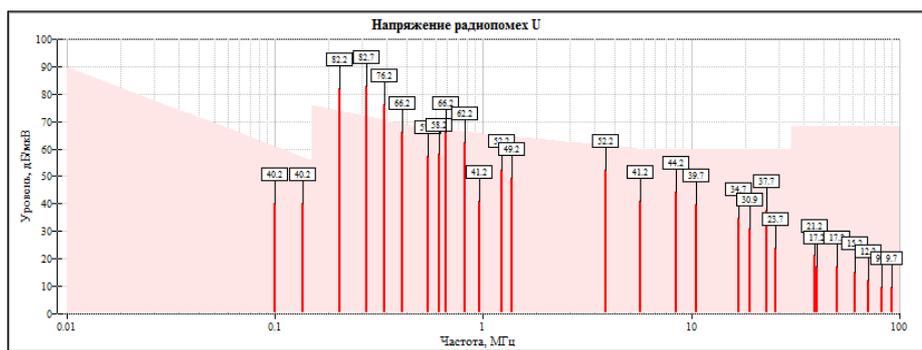


Рис. 1 Напряжения промышленных радиопомех опытного образца ЭВМ.

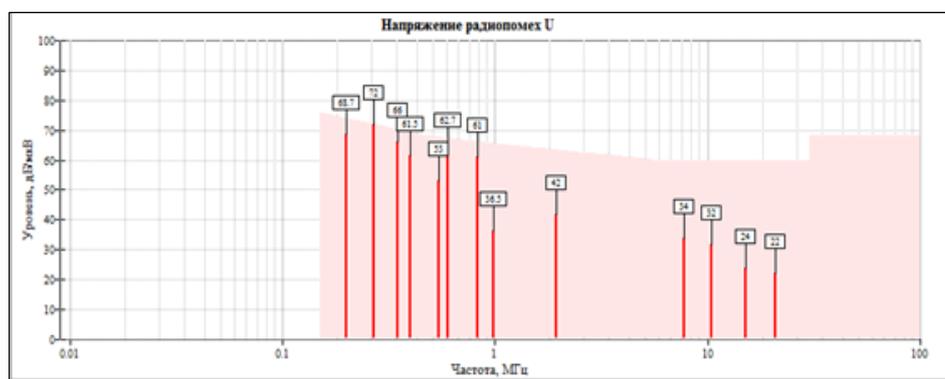


Рис. 2 Напряжения промышленных радиопомех опытного образца ЭВМ после доработки фильтра питания

Таблица 3. Напряжения промышленных радиопомех опытного образца ЭВМ после доработки фильтра питания.

$F_i$ , МГц	$U_{изм}$ , дБ мкВ	$K$ , дБ	$U_{пом}$ , дБ мкВ	$U_{норм}$ , дБ мкВ	Превыше- ние, дБ мкВ
0.20	68.70	0.00	68.70	72.60	-
0.27	72.00	0.00	72.00	71.88	0.12
0.35	66.00	0.00	66.00	71.39	-
0.40	61.50	0.00	61.50	68.70	-
0.55	53.00	0.00	53.00	65.97	-
0.60	62.70	0.00	62.70	66.22	-
0.83	61.00	0.00	61.00	67.24	-
0.98	36.50	0.00	36.50	67.91	-
1.94	42.00	0.00	42.00	64.35	-
7.71	34.00	0.00	34.00	60.00	-
10.48	32.00	0.00	32.00	60.00	-
15.16	24.00	0.00	24.00	60.00	-
20.75	22.00	0.00	22.00	60.00	-

Проводились измерения напряжения промышленных радиопомех опытного образца ЭВМ после доработки фильтра питания. Испытания проводились на альтернативной измерительной площадке ФГУ ФНЦ НИИСИ РАН.

Результаты измерения в области нижних частот, в которой наблюдалось превышение допустимых уровней радиопомех, приведены ниже в таблице 4 и на рисунке 2.

Изменение параметров затухания фильтра опытного образца ЭВМ в области низких частот позволило уменьшить уровни напряжения промышленных радиопомех до допустимых.

## 5. Заключение

Предложены методики и общие принципы конструирования электронных модулей OpenVPX для защищенных серверов, методики уменьшения помех в корпусах ЭВМ в стойечном исполнении.

Проведены измерения на макете ЭВМ. Для ЭВМ с терафлопсной производительностью выбрана конструкция и методы защиты от ЭМП.

Для перспективных ЭВМ (процессор Базис-Б1) применение методик позволит конструировать сервера с увеличенной производительностью с подобными габаритно-весовыми характеристиками.

## Источники финансирования

Публикация выполнена в рамках выполнения госзадания НИР «Разработка общих принципов и методик создания корпусов моноблоков и серверов в соответствии с требованиями по уровням электромагнитных излучений и отводу тепла» (0065-2018-0018).

# Methods of limiting electromagnetic interference of electronic modules and server type ECM

A.S. Bakulin, S.A. Bakhov, O.V. Serdin, A.G. Prozorova, I.V. Tokarev

**Abstract:** The report presents an analysis of main methods used to limit EMI. The methods and general principles of designing electronic modules for secure servers are proposed. This paper also concludes methods for reducing interference in rack-type computers. The effectiveness of the proposed methods is confirmed by experimental data obtained at the alternative measuring site SRISA.

**Keywords:** secure server housing, mechanisms of limiting electromagnetic interference, domestic dual-core 64-bit microprocessor, air-cooling.

## Литература

1. T.Mark, K.Motohisa, L.Myron, L.Crawford, Ezra B. Larsen. A review of electromagnetic compatibility/interference measurement methodologies. Proceedings of the IEEE, vol.73, no.3, 388-411.
2. A. Hesener. Electromagnetic interference (EMI) in power supplies. «Fairchild semiconductor power seminar 2010-2011».
3. S.Braun, P.Russer. Uncertainty analysis and novel test procedures performed with a real-time time-domain EMI measurements system. «2007 IEEE International symposium on electromagnetic compatibility».
4. Н.В. Балюк, А.А. Давыдов, Д.И. Крохалев, С.Д. Орлов. Расчетно-экспериментальные методы оценки соответствия технических средств заданным требованиям электромагнитной стойкости и безопасности. «Технологии ЭМС», т.2 (2018), 7-22.
5. В.А. Михеев. Расчет показателей эффективности электромагнитной безопасности многофункциональной информационно системы интегрированной структуры. «Технологии ЭМС», т.1 (2017), 47-51.

# Методика разработки универсального керамического корпуса микросхемы

А.А. Подковыров

ФГУ ФНЦ НИИСИ РАН, Москва, Россия, E-mail: [barfey@cs.niisi.ras.ru](mailto:barfey@cs.niisi.ras.ru)

**Аннотация:** В статье рассмотрена методика создания универсального корпуса на примере разработки корпуса для изготовления двух микросхем: процессора и контроллера. Дана оценка эффективности рассмотренной методики и указаны ее преимущества. Работа была выполнена на предприятии ФГУ ФНЦ НИИСИ РАН.

**Ключевые слова:** корпусирование, керамический корпус, универсальный корпус, микросхема.

## Введение

Практически каждая оптимизация разработки корпуса включает в себя проработку возможности совмещения нескольких микросхем в одном корпусе. Разработка корпуса и, как минимум, расстановка выводов кристалла происходят параллельно друг другу, между разработчиками ведется постоянный обмен информацией. Именно в процессе расстановки выводов можно сделать заключение, возможно ли совмещение нескольких кристаллов в один корпус.

Разработанная автором методика позволяет решить две задачи, связанные с разработкой универсального корпуса для нескольких микросхем: оценить возможность совмещения нескольких кристаллов в один корпус и свести разработку универсального корпуса к упрощенной разработке топологии кристалл – корпус. В результате применения этой методики процесс разработки сводится к двум этапам: оценка совместимости и сведение к разработке кристалл – корпус.

Методика была использована при разработке радиационно-стойкого керамического корпуса для двух микросхем: процессора со встроенным сопроцессором и контроллера состояния модулей VPX на предприятии ФГУ ФНЦ НИИСИ РАН.

## Этап оценки совместимости

Как сложность такой разработки, так и сама возможность совмещения двух и более кристаллов зависят от нескольких

основных факторов. Реализация требований каждого последующего фактора более трудоемкая и, в некоторых случаях, требует переработки предыдущих оптимизаций. Первый и наиболее значимый фактор – это степень различия в составе интерфейсов во всех вариантах микросхем.

Невозможно совместить абсолютно отличные друг от друга микросхемы, это не эффективно в принципе. Вторым фактором является общее количество выводов микросхемы, включающее в себя распределение между сигнальными выводами и выводами питания/земли. Обычно стоит задача минимизировать количество используемых выводов, но, если выводов микросхемы для размещения интерфейсов различных микросхем на одной и те же выводы не хватает, то третьим фактором становится проработка совмещения интерфейсов внутри определенной группы выводов. В разных микросхемах на этой группе выводов будут расположены различные интерфейсы. На основании проработки всех факторов и, соответственно, расстановки всех интерфейсов, можно сделать заключение о возможности совмещения нескольких кристаллов в один корпус. На основании этого заключения принимается решение о целесообразности дальнейшей разработки универсального корпуса.

## Описание микросхем

Для решения задачи расстановки выводов проведем анализ состава интерфейсов микросхем.

В начале рассмотрим микросхему процессора со встроенным сопроцессором. Перечень интерфейсов представлен ниже:

- DDR3 (235 выводов)
- 4 канала SRIO (160 выводов)
- 2 канала MKIO (16 выводов)
- 6 каналов SPI (41 вывод)
- 2 канала SpaceWare (46 выводов)
- 2 канала UART (8 сигналов)
- 2 канала CAN (10 выводов)
- 2 канала I2C (4 вывода)
- RGMII (15 выводов)
- DBus(staticSRAM/PROM)(84 вывода)
- GPIO A, B (32 вывода)
- JTAG (5 выводов)
- Служебные сигналы (35 выводов)

Итого 691 вывод (без питания/земли ядра микросхемы)

Данный процессор предполагалось разместить в корпусе размером 30 на 30 выводов. Процессор малопотребляющий и по результатам моделирования порядка 200 выводов на питания/земли ему достаточно.

Микросхема контроллера содержит следующие интерфейсы:

- DDR2 (240 выводов)
- 4 канала SRIO (160 выводов)
- MKIO (8 выводов)
- 2 канала SPI (14 выводов)
- 2 канала SpaceWare, SW (46 выводов)
- 2 канала UART (8 выводов)
- 2 канала CAN (10 выводов)
- 4 канала I2C (8 выводов)
- MII (25 выводов)
- 4 канала DBUS DIFF (96 выводов)
- GPIO A, B (32 вывода)
- SRAM (54 выводов)
- ONFI (21 вывод)
- JTAG (5 выводов)
- Служебные сигналы (20 выводов)

Итого 747 выводов (без учета питания/земли ядра микросхемы)

Контроллер имеет большое количество блоков, которые работают с периферией. К тому же, требуется одновременная работа этих блоков. Разработчики блоков кристалла, на основе их анализа, передали информацию, что

только 24 старших вывода GPIO могут быть объединены со старшими выводами данных SRAM. В итоге получено 723 вывода. Оставшихся 175 выводов, выделенных для питания/земли микросхемы, по результатам моделирования, достаточно для работы микросхемы.

## Оценка факторов и расстановка выводов

Итак, сами микросхемы по отдельности укладываются в выбранный керамический корпус с 898 выводами. По описанию выводов микросхем, в результате подсчета, видно, что размещение интерфейсов, особенно в микросхеме контроллера, очень плотное, поэтому невозможно выделить отдельные выводы для отличных друг от друга интерфейсов. Перейдем к сравнению по составу интерфейсов этих двух микросхем.

В результате проведения совмещения по интерфейсам было установлено что интерфейсы, перечисленные ниже, могут полностью совпадать в универсальном корпусе (в скобках указано общее количество занимаемых выводов):

- DDR (240 выводов)
- 4 канала SRIO (160 выводов)
- 1 канал MKIO (8 выводов)
- 2 канала SPI (14 выводов)
- 2 канала SpaceWare, SW (46 выводов)
- 2 канала UART (8 сигналов)
- 2 канала CAN (10 выводов)
- 2 канала I2C (4 вывода)
- JTAG (5 выводов)

Итого 495 выводов.

По информации от разработчиков кристалла, блоки DDR3 не имеют свободного размещения выводов, выводы в блоке DDR2 могут свободно размещаться. Поэтому за основу размещения выводов на совмещенном корпусе взята расстановка выводов блока DDR3. На несовпадающие по назначению сигналы и добавочные 5 выводов, были назначены оставшиеся выводы DDR2. В итоге, несмотря на отличие интерфейсов DDR2 и DDR3, интерфейс DDR2 с 240 выводами совмещается с интерфейсом DDR3. По интерфейсам MKIO, SPI и I2C микросхемы не совпадают только

по количеству каналов, поэтому выделяем то количество каналов, которое присутствует в обоих вариантах микросхем: 1 канал МКЮ, 2 канала SPI и 2 канала I2C. Остальные интерфейсы, представленные в списке, совпадают по всем параметрам, и их проработка заключается только в 2 действиях: географически правильном расположении на модуле и оптимизации расстановки выводов для трассировки на печатной плате, что относится к этапу оптимизации внутри микросхемы.

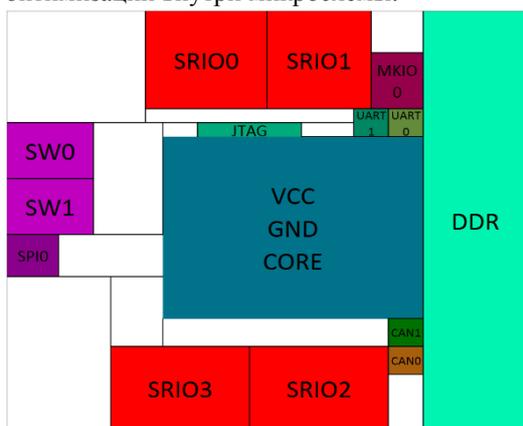


Рисунок 1. Совпадающие интерфейсы на универсальном корпусе

В итоге, исходя из расположения интерфейсов на применяемых для этих микросхем модулях, и из-за возможности их размещения на кристаллах, эти интерфейсы географически располагаются как показано на рисунке 1 (каналы I2C не представлены, т.к. этот интерфейс не требовательный к размещению и включает всего 2 сигнала; он будет размещен в последнюю очередь).

Третий, заключительный фактор в расстановке интерфейсов – это проработка совмещения интерфейсов внутри определенной группы выводов.

В разных микросхемах на этой группе выводов могут находиться различные интерфейсы.

Составим список интерфейсов для каждой микросхемы, располагая их от наиболее требовательных высокоскоростных интерфейсов до самых простых по требованиям к их расположению и трассировке по корпусу.

Интерфейсы процессора и контроллера представлены таблицами 1 и 2 соответственно:

Таблица 1. Интерфейсы микросхемы процессора для совмещения

Интерфейс	Количество выводов	Требования
1 канал МКЮ	16	Дифференциальные пары без разрывов опорных полигонов
RGMII	15	Сигнальные линии без разрывов опорных полигонов
DBus (static SRAM/PROM)	84	Сигнальные линии без разрывов опорных полигонов
GPIO A, B	32	Без особых требований
4 канала SPI	41	Без особых требований

Таблица 2. Интерфейсы микросхемы контроллера для совмещения

Интерфейс	Количество выводов	Требования
4 канала DBUS DIFF	96	Дифференциальные пары без разрывов опорных полигонов
SRAM	54	Сигнальные линии без разрывов опорных полигонов
MII	25	Желательно сигнальные линии без разрывов опорных полигонов
ONFI	21	Желательно сигнальные линии без разрывов опорных полигонов
GPIO A	8	Без особых требований
2 канала I2C	8	Без особых требований

В соответствии со сложностью реализации требований размещения, начинаем расстановку с наиболее требовательных интерфейсов. Если таковых несколько, то с интерфейсом, который имеет наибольшее количество выводов. В нашем случае, этим интерфейсом является 4 канальный DBUS DIFF. Как для дифференциальных пар, да и для всех требовательных сигналов наилучшим расположением является их расположение ближе к краю корпуса, т.к. их необходимо трассировать в первую очередь. В существующей предварительной расстановке таких мест два: слева сверху и слева снизу. Исходя из расположения интерфейса на модуле, расположение сверху слева оптимально.

Далее по сложности следует интерфейс на процессоре DBUS (static SRAM/ PROM). Интерфейс SRAM на микросхеме контроллера является частным случаем DBUS (static SRAM/ PROM), поэтому, логичным выглядит их совмещение путем размещения на корпусе изначально полного варианта. Подходящее место для этого – левый нижний угол. На оставшиеся выводы микросхемы контроллера установим 8 младших битов GPIO A, так как старшие биты GPIO совмещены с SRAM интерфейсом.

Интерфейсы MII, RGMII и ONFI разместим исходя из требований их расположения на модуле. RGMII необходимо разместить в левом верхнем углу. RGMII интерфейс принадлежит только

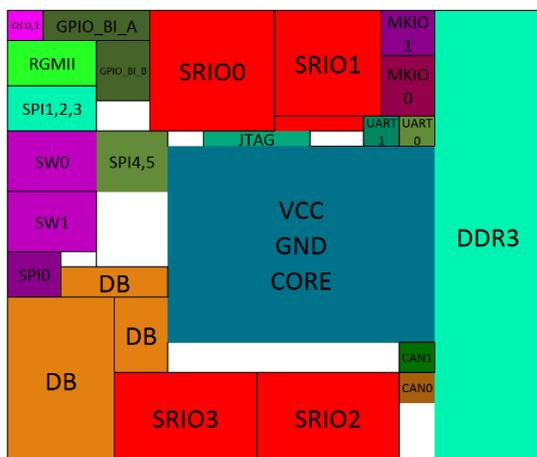


Рисунок 2. Расположение выводов на корпусе процессора

процессору, а расположенный на совмещенном корпусе интерфейс 4 канальный DBUS DIFF только контроллеру. Осталось только получить подтверждение, что такая расстановка реализуема в кристалле, т.к. на то же место блока интерфейса 4 канального DBUS DIFF должен будет установлен RGMII. По утверждению разработчиков кристалла, выводы на кристалле интерфейса RGMII могут там разместиться. Также предложено на оставшиеся выводы 4 канального интерфейса DBUS DIFF разместить свободно перемещаемые интерфейсы GPIO, I2C и часть каналов интерфейса SPI. Исходя из расположения интерфейсов на модуле, интерфейс MII нужно разместить за интерфейсами SRIO2,3, а нетребовательный интерфейс ONFI разместить, разделив на две части, одну часть за интерфейсами SRIO0,1, вторую часть за MKIO. На кристалле этот блок будет находиться на одном месте, а соответствующие части интерфейса будут трассироваться на соответствующие выводы на корпусе.

После расстановки требовательных интерфейсов и интерфейсов, расположение которых задано их географическим расположением на модуле, оставшиеся 2 канала I2C контроллера, по согласованию с разработчиками кристаллов, разместились на свободные выводы процессорного DBUS (static SRAM/ PROM).

Итоговое расположение выводов на процессоре и контроллере показано на рисунке 2 и 3 соответственно.

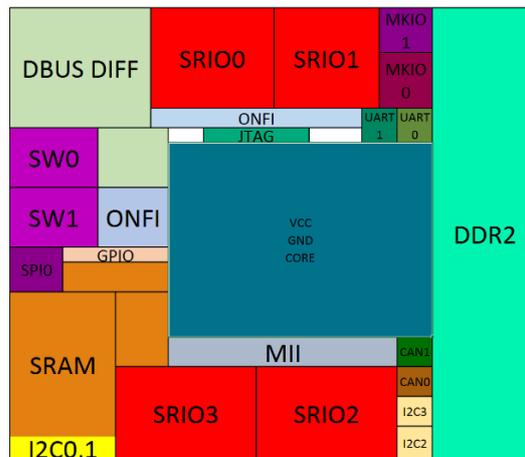


Рисунок 3. Расположение выводов на корпусе контроллера

## Разработка универсального корпуса

В соответствии с методикой разработка универсального корпуса делится на несколько стадий:

1. составление списка для замены/объединения цепей в каждом кристалле.
2. автоматическая замена/объединение цепей в каждом кристалле.
3. автоматическая сверка двух кристаллов.
4. трассировка полностью совпадающих интерфейсов.
5. трассировка совмещенных интерфейсов.
6. доработки корпуса с учетом моделирования.

Рассмотрим каждый из них более подробно.

### Составление списка для замены/объединения цепей

Исходя из опыта проектирования корпуса, даже одного кристалла, невозможно сделать полностью идентичными названия цепей и в кристалле, и в корпусе, тем более,

есть необходимость объединения нескольких цепей именно в корпусе. В случае с универсальным корпусом задача усложняется, один и тот же вывод на разных кристаллах может иметь не только отличаться названием одного и того же сигнала, но могут принадлежать к сигналам разных интерфейсов. В любом случае, без списка замены названия сигналов с какой-либо из сторон, либо со стороны корпуса, либо со стороны кристалла, не обойтись. Так как корпус один, а кристаллов два, корректнее заменять названия в кристаллах, приводя их к общему для обоих корпусов названию, которое будет содержать в своем имени информацию обо всех его назначениях.

Для особенных сигналов типа дифференциальных пар или сигналов, которые совмещены, итоговые названия будут сложными и длинными. Их названия рекомендуется задавать, основываясь на определенной структуре. Рассмотрим наиболее распространенные варианты названий в таблице 3:

Таблица 3. Примеры различных вариантов названий сигналов

Первая микросхема	Вторая микросхема	Совмещенное название
MKIO1_ENR	MII_TXD[1]	MKIO1_ENR_I_MII_TXD1
NC	DDR_DUMIN	NC_I_DDR0_DUMIN
MKIO1_DMI_P	SPI1_MISO	MKIO1_DMI_I_SPI1_MOSIM_SPI1_MISOP_P
MKIO1_DMI_M	SPI1_MOSI	MKIO1_DMI_I_SPI1_MOSIM_SPI1_MISOP_M
DDR_MCLK3_M	DDR_DQS_M[9]	DDR_MCLK3_I_DDR_DQSM9_M
DDR_MCLK3_P	DDR_DQS_P[9]	DDR_MCLK3_I_DDR_DQSM9_P
GPIO_D[5]	DBL_CLKI_N[3]	GPIO5M_SPI3_CS0NP_I_DBUS3_CLKI_M
SPI3_CS[0]_N	DBL_CLKI_P[3]	GPIO5M_SPI3_CS0NP_I_DBUS3_CLKI_P

### Автоматическая замена/объединение цепей

Количество итераций при разработке одного кристалла может достигать нескольких десятков, поэтому этот процесс рекомендуется автоматизировать. Автором была разработана программа «Replace Names» которая, получив на входе список

исходных сигналов и список замены, выдавала результат, в том числе в удобном для использования формате, для вставки в приложение Microsoft Excel. Использование программы позволило исключить ручную обработку входной информации, тем самым сократить ввод обновленной информации и минимизировать количество ошибок при замене и объединении цепей. Внешний вид приложения представлен рисунке 4.

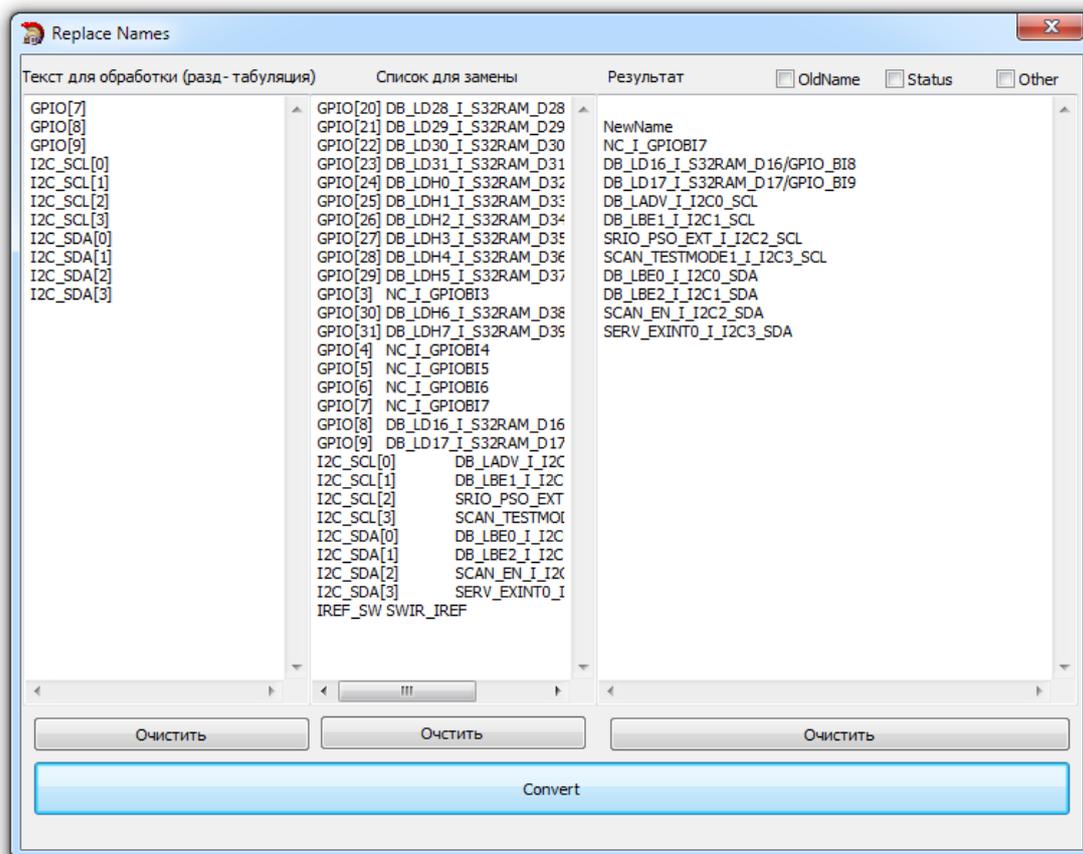


Рисунок 4. Программа «Replace Names», автозамена названия сигналов

## Автоматическая сверка двух кристаллов

Как уже писалось ранее, во введении, корпус и кристалл (а в нашем случае 2 кристалла) разрабатываются параллельно, поэтому количество итераций при разработке, в частности операций совмещения кристалла и корпуса, очень значительно. Чтобы внесение изменений было максимально быстрым и не содержало ошибок, необходимо этот процесс автоматизировать. Предлагаемый процесс проходит в два шага: сначала исходные данные автоматически преобразуются с помощью программы «Replace Names» и списка замены, а затем эти данные совмещаются в программе Microsoft Excel. После замены, названия цепей в кристаллах должны совпадать. Пример на рисунке 5.

Количество операций в этой таблице может быть расширено. В нее могут быть

включены и сравниваться координаты контактных площадок, их названия, направления и типы сигналов. Кроме того, в этой таблице можно провести сравнение результатов разных итераций (столбик old result crist на рисунке 5).

Благодаря всем вышеописанным операциям сверки кристаллов почти полностью исключаются следующие ошибки:

- Замыкание цепей земли и питания в разных кристаллах (в одном кристалле вывод земля, в другом питание).
- Нарушение целостности вывода: координат, свойств, названия и типа.
- Нарушения при совмещении выводов – назначение вывода не совпадает в разных кристаллах.
- Ошибки списка замены: назначение вывода имеет ошибки в слове для замены или оно не совпадает с исходным названием в кристалле.

Q	R	S	T	U
1 crist	2 crist	result crist	old result crist	срав стар ио7
MII_MDC	CAN0_M	CAN0_I_MII_MDCM_MII_RXERRP_M	CAN0_I_MII_MDCM_MII_RXERRP_M	ИСТИНА
MII_RXERR	CAN0_P	CAN0_I_MII_MDCM_MII_RXERRP_P	CAN0_I_MII_MDCM_MII_RXERRP_P	ИСТИНА
CAN_EN[0]	CAN0_RS	CAN0_RS	CAN0_RS	ИСТИНА
CAN_RX[0]	CAN0_RX	CAN0_RXD	CAN0_RXD	ИСТИНА
CAN_TX[0]	CAN0_TX	CAN0_TXD	CAN0_TXD	ИСТИНА
MII_MDI0	CAN1_M	CAN1_I_MII_MDIOM_MII_CRSP_M	CAN1_I_MII_MDIOM_MII_CRSP_M	ИСТИНА
MII_CRSP	CAN1_P	CAN1_I_MII_MDIOM_MII_CRSP_P	CAN1_I_MII_MDIOM_MII_CRSP_P	ИСТИНА
CAN_EN[1]	CAN1_RS	CAN1_RS	CAN1_RS	ИСТИНА
CAN_RX[1]	CAN1_RX	CAN1_RXD	CAN1_RXD	ИСТИНА
CAN_TX[1]	CAN1_TX	CAN1_TXD	CAN1_TXD	ИСТИНА
GEP[3]	CP2_PSO_EXT	CP2_PSO_EXT_I_GEP3	CP2_PSO_EXT_I_GEP3	ИСТИНА
MEM_ADR[0]	DB_LA[0]	DB_LA0_I_S32RAM_A0	DB_LA0_I_S32RAM_A0	ИСТИНА
MEM_ADR[1]	DB_LA[1]	DB_LA1_I_S32RAM_A1	DB_LA1_I_S32RAM_A1	ИСТИНА
MEM_ADR[10]	DB_LA[10]	DB_LA10_I_S32RAM_A10	DB_LA10_I_S32RAM_A10	ИСТИНА
MEM_ADR[11]	DB_LA[11]	DB_LA11_I_S32RAM_A11	DB_LA11_I_S32RAM_A11	ИСТИНА
MEM_ADR[12]	DB_LA[12]	DB_LA12_I_S32RAM_A12	DB_LA12_I_S32RAM_A12	ИСТИНА
MEM_ADR[13]	DB_LA[13]	DB_LA13_I_S32RAM_A13	DB_LA13_I_S32RAM_A13	ИСТИНА
MEM_ADR[14]	DB_LA[14]	DB_LA14_I_S32RAM_A14	DB_LA14_I_S32RAM_A14	ИСТИНА
MEM_ADR[15]	DB_LA[15]	DB_LA15_I_S32RAM_A15	DB_LA15_I_S32RAM_A15	ИСТИНА

Рисунок 5. Пример автоматического сравнение цепей в программе Microsoft Excel

## Трассировка полностью совпадающих интерфейсов

Для полностью совпадающих интерфейсов разработка топологии ничем не отличается от трассировки обычного корпуса для одного кристалла. Разработка производится в соответствии с требованиями к трассировке этих интерфейсов. Рекомендуется разрабатывать сначала интерфейс, который имеет наиболее жесткие требования, так как для трассировки можно будет использовать все пространство корпуса. Далее разрабатываются интерфейсы по убыванию их сложности, их легче вписать в корпус с частично занятым пространством. Интерфейсы без особых требований можно будет разместить по остаточному принципу, возможно даже не группируя их в одном месте.

## Трассировка совмещенных интерфейсов

После совмещения расстановки выводов на корпусе и на кристалле, трассировка также мало отличается от трассировки обычного корпуса для одного кристалла, за исключением некоторых

особенностей. Сама возможность совмещения интерфейсов прорабатывалась на этапе расстановки, поэтому при трассировке совмещенных интерфейсов нужно учитывать только лишь требования для обоих интерфейсов. Если интерфейсы имеют разные требования к топологии, то за основу берутся наиболее жесткие требования, по аналогии с разработкой топологии для полностью совпадающих интерфейсов, за исключением того, что требования складываются, то есть необходимо соблюдение всех требований.

## Доработки корпуса с учетом моделирования

После завершения разработки топологии корпуса и проверки его по чек-листу, файл, который содержит все данные о топологии корпуса, передается на моделирование. На этапе моделирования проверяется соблюдение требований по нагрузочной способности цепей питания, удовлетворение требований по интерфейсам, проводится проверка кросс-наводок сигнальных линий, а также производится моделирование с учетом установленных моделей кристаллов.

В результате выполнения указанных выше проверок, создаются отчеты. По информации из этих отчетов корректируется топология корпуса. Если в результате моделирования требования каких-либо интерфейсов не удовлетворяются, то корпус дорабатывается и вновь отправляется на моделирование. И так до выполнения всех требований.

## Заключение

В статье рассмотрена методика разработки универсального керамического корпуса для установки в него одного из двух кристаллов, для изготовления двух различных микросхем: микросхемы процессора со встроенным сопроцессором и микросхемы контроллера состояния модулей. Благодаря применению этой методики удалось сократить время разработки в полтора раза по сравнению с временем разработки двух корпусов с одним кристаллом в каждом. Время разработки складывалось из продолжительности двух этапов: сведение двух кристаллов и двух корпусов к одному кристаллу и одному

корпусу, и стандартная разработка топологии кристалл – корпус. Первый этап занял примерно половину времени, которое потребовалось для выполнения второго этапа. Учет всех фактов при расстановке выводов на корпусе позволяет сделать расстановку интерфейсов за одну - две итерации для большинства корпусов. Автоматизация процесса подготовки совмещенного кристалла позволила увеличить скорость внесения изменений в топологию корпуса и исключить множество ошибок, в том числе найти ошибки при совмещении выводов двух кристаллов в один.

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (выполнение фундаментальных научных исследований ГП 14) по теме № 0065-2018-0009 "Архитектурные и схемотехнические методы снижения энергопотребления и повышения сбоеустойчивости микропроцессоров и коммуникационных контроллеров высокопроизводительных ЭВМ." (№АААА-А18-118041190168-0).

# The method of development of universal ceramic substrate

**A.A.Podkovyrov**

**Abstract:** The article describes the method of creating a chip substrate on the example of the development a substrate for two chips: processor and controller. An assessment of the effectiveness of the considered methodology is given and its advantages indicated. The work was performed at the enterprise SRISA.

**Keywords:** Flip Chip Packaging, Ceramic Substrate, Universal Substrate, Flip Chip BGA.

## Литература

1. A.B.Chong, H.K.Chun. Unified Padring Design Flow // AIMS'13 / Proceedings of the 2013 1st International Conference on Artificial Intelligence, Modelling and Simulation, pp. 462-466.
2. R.J. Lee, M.F. Lai, and H.M. Chen. Fast flipchip pinout designation respin by pinblock design and floorplanning for package-board codesign // Design Automation Conference, 2007 /ASPDAC'07, Jan. 2007, pp. 804-809.

# Повышение чувствительности установки контроля дефектов в светлом поле при наличии вариаций яркости изображения

А.А.Захарченко<sup>1</sup>, Е.П.Михальцов<sup>2</sup>, М.В.Орешков<sup>3</sup>, В.Ю.Троицкий<sup>4</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

E-mail's: <sup>1</sup> [Zakh\\_Stu@rambler.ru](mailto:Zakh_Stu@rambler.ru), <sup>2</sup> [Evgeniy\\_Mikhaltsov@srisa.ru](mailto:Evgeniy_Mikhaltsov@srisa.ru)

<sup>3</sup> [oreshkovmv@mail.ru](mailto:oreshkovmv@mail.ru), <sup>4</sup> [Viacheslav\\_Troitskiy@srisa.ru](mailto:Viacheslav_Troitskiy@srisa.ru)

**Аннотация:** В статье описан метод повышения чувствительности установки светлопольного контроля дефектности на полупроводниковых пластинах с топологией, при наличии вариаций яркости изображения по пластине, обусловленных присутствием тонких диэлектрических слоев различной толщины. Показана реализация этого метода на основе установки ЭМ-6429Б в ФГУ ФНЦ НИИСИ РАН.

**Ключевые слова:** контроль дефектности, топология, светлопольный, вариации яркости

## 1. Введение

Метод контроля дефектности на кремниевых пластинах с топологическим рисунком в светлом поле (brightfield inspection) широко применяется в микроэлектронном производстве во всём мире. Он заключается в последовательном сканировании поверхности пластины с помощью видеокамеры при освещении в светлом поле и в последующем поиске отличий в изображениях соседних кристаллов с помощью специализированного программного обеспечения по обработке изображений. При сканировании задается порог яркости, при превышении которого отличие в изображениях считается дефектом.

Существуют несколько компаний, производящих данный тип оборудования, такие как KLA-Tencor, Applied Materials (обе США) и др. Аналогичная установка ЭМ-6429Б была впервые изготовлена на предприятии КБТЭМ-ОМО, Республика Беларусь. Установка успешно прошла испытания и в настоящее время используется для контроля технологических процессов в кристалльном производстве при изготовлении микросхем в ФГУ ФНЦ НИИСИ РАН.

Для контроля дефектности на пластинах на установке ЭМ-6429Б разработчиками было решено использовать светодиодный осветитель, который обладает неоспоримыми преимуществами перед лампой накаливания и ртутной лампой: большой срок службы (30 000 ч. по сравнению с 2000 ч.), малое

энергопотребление и малое тепловыделение, что положительно влияет на стабильность работы оптической системы установки.

Однако, из всех используемых типов освещения, светодиод имеет самый узкий спектр, что приводит к возникновению сильных вариаций яркости изображения тонких полупрозрачных пленок различной толщины [1].

На рисунках 1, а-в показаны изображения одного и того же участка микросхемы в соседних кристаллах пластины, полученные на этапе контроля процессов формирования целевой изоляции. Светлые области на левом рисунке – это области целевой изоляции - канавки в кремнии глубиной порядка 5000 Å, заполненные окислом. Эти области имеют практически постоянную яркость на всех кристаллах.

Для различных структур яркость их изображений в данной установке варьируется в интервале от 0 до 255 относительных единиц (о.е.). Для представленных изображений она составляет 137 о.е. Отличия на изображениях лишь кажущиеся из-за особенностей восприятия изображения человеческим глазом.

Остальные области – это «островки» кремния, покрытые окислом толщиной 100-200 Å, который дает сильные вариации яркости, в данном случае от 87 о.е до 143 о.е. На рисунке 1, г представлена разница яркости изображений, показанных на рис. 1, а и в, для наглядности усиленная в 3 раза.

Видно, что дефект, присутствующий на рис. 1, а (показан стрелкой), не может быть найден, т.к. его отличие по яркости (9 о.е.)

существенно меньше отличия по яркости других областей изображения, достигающего 30 о.е.

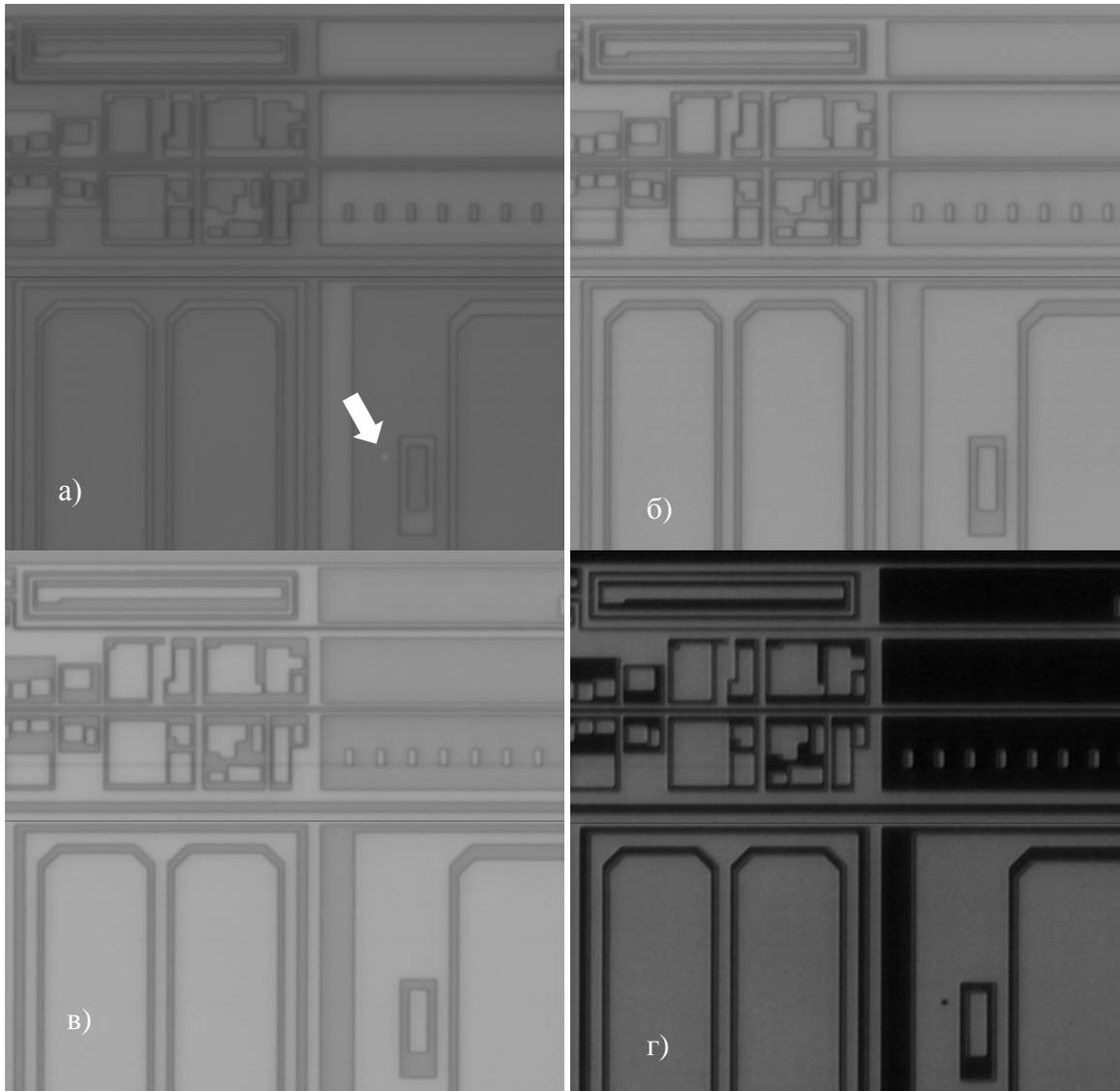


Рис. 1. (а), (б), (в) - изображения одного и того же участка микросхемы в соседних кристаллах пластины; (г) - разница между первым и третьим изображением, усиленная в 3 раза.

На установках фирмы KLA-Tencor данная проблема решается использованием освещения широкого спектра (Broadband Illumination) [2,3], который захватывает ультрафиолетовую область. Однако даже при таком широком спектре освещения спектр сигнала, регистрируемый камерой ограничивается её спектральной чувствительностью, и вариации яркости все равно присутствуют, хотя и в меньшей степени. Нашей же задачей являлось исследование возможности повышения чувствительности установки в случае светодиодного освещения.

В данной статье описывается алгоритм повышения чувствительности установки при наличии вариаций яркости изображений.

## 2. Постановка задачи

Для решения задачи поиска дефектов топологического рисунка в режиме светлого поля должно быть просканировано  $N \geq 3$  кристаллов, изображения которых будут затем сравниваться между собой. Пусть  $I(x, y)$  – распределение яркости на изображении (сигнал) отсканированного участка пластины размером  $(K * N, H)$ , где  $K$  – размер кристалла,

$H$  – ширина полосы сканирования, не превышающая размер кристалла.

Тогда изображение (эталонное) можно разложить на постоянную часть и меняющуюся:

$$I(x, y) = f(x \div K, y)\chi_i(x \div K, y) + c(x)(1 - \chi_i(x \div K, y)) + v(x, y), \quad (1)$$

где операция " $\div$ " – это остаток от деления;

$\chi_i$  – функция, равная 1, если точка лежит в постоянной области, и 0, если точка в меняющейся области;

$c(x)$  – распределение яркости пленки по пластине (можно считать что, для узкой полосы сканирования,  $K * N/H \sim 20 - 100$ , оно зависит только от  $x$ ),

$f(x, y)$  – постоянная часть изображения,

$v(x, y)$  – шум, случайная величина с нулевым математическим ожиданием и дисперсией  $\sigma^2$ .

Если мы определим функции  $\chi_i(x, y)$  и  $c(x)$ , то сможем анализировать сигнал:

$$\tilde{I}(x, y) = I(x, y) - c(x)(1 - \chi_i(x \div K, y)), \quad (1.)$$

который будет избавлен от вариаций яркости.

Заметим также, что на функцию  $c(x)$  накладываются условия непрерывности и «гладкости», обусловленные физическим смыслом. Условие «гладкости» заключается, например, в ограничении на производную  $|c'(x)| < d$ , где  $d$  – некоторая константа.

Задача поиска  $\chi_i(x, y)$  и  $c(x)$  не линейна и решается лишь численными методами приближенно. Так как размер получаемого изображения может достигать 1 Гигапикселя, для ускорения решения задачи предлагается разбить ее на две части: поиск функции  $\chi(x, y)$ , и поиск  $c(x)$ .

Для точек кристаллов с координатами  $(x, y)$ , где  $x < K, y < H$ , из областей, не

подверженных вариациям яркости, набор чисел

$$\{Y_i(x, y), i = 1..N\}; Y_i(x, y) = I(x + K \times i)$$

будет либо содержать примерно одни и те же числа, либо иметь одно сильно отличающееся от остальных число, если в этой точке кристалла находится дефект (вероятность наличия двух дефектов в одной и той же точке крайне мала). Для точек из областей с вариациями яркости это будут наборы различных чисел.

Задачу отнесения множества  $\{Y_i(x, y)\}$  к набору «почти» одинаковых чисел можно решить с помощью теории проверки статистических гипотез, заметив, что  $Y_i(x, y)$  является случайной величиной, т.к. в нашей модели (1.) при  $\chi_i(x, y) = 1, I(x + K \times i, y) - f(x, y) = v(x + K \times i, y)$ .

Поэтому гипотеза для решения  $\chi_i(x, y) = 1$  состоит в следующем: величина  $Y_i(x, y)$  при исключении одной любой точки имеет распределение с дисперсией не более  $\sigma^2$ , при альтернативе, заключающейся в том, что дисперсия распределения больше  $\sigma^2$ . Здесь применено условие «больше», т.к. дисперсия шума, как правило, не известна, поэтому лучше ввести ограничение снизу. К тому же при таком условии задача решается быстрее простым исключением максимальной или минимальной «выпадающей» точки из набора и отсечением порогом на стандартное отклонение набора  $\{Y_i(x, y)\}$ . Пороговое значение можно выбрать экспериментально по изображению, т.к. оно может быть различным для каждого конкретного случая. Для этого следует построить гистограмму дисперсий наборов  $\{Y_i(x, y)\}$  без «выпадающей точки» (пример показан на рис. 2), распределение должно быть мультимодальным, в противном случае исследуемое изображение не вписывается в нашу модель (1).

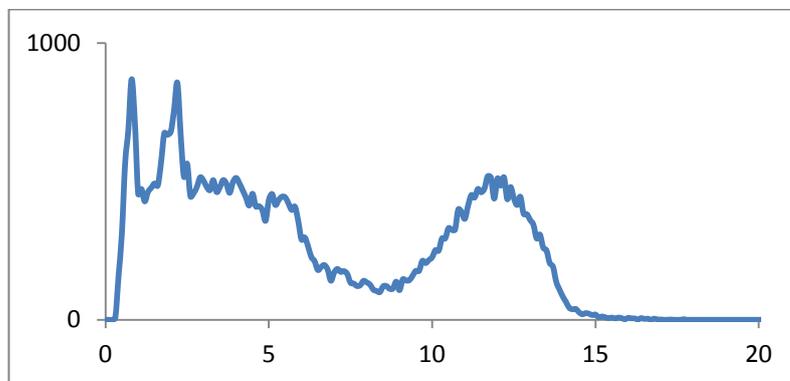


Рис 2. Гистограмма распределения корня из дисперсии наборов чисел без выпадающей точки по 400 000 точкам в кристалле.

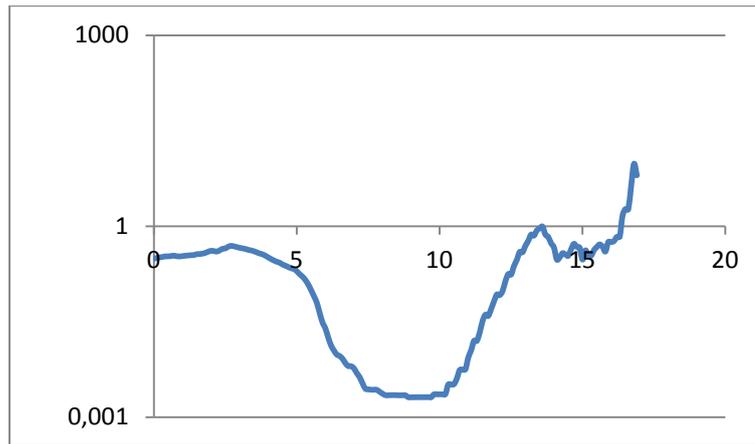


Рис 3. График функционала для поиска максимальной моды.

Отделение максимальной моды, которая соответствует вариативной части изображения, осуществляется с помощью метода проверки на унимодальность, описанного в [4,5].

Метод основан на построении проекции на множество конусов и состоит в

поиске минимума функционала равного отношению расстояния вектора до множества конусов и его расстояния до константы.

Пример такого функционала для распределения из рис. 2 показан на рис. 3.

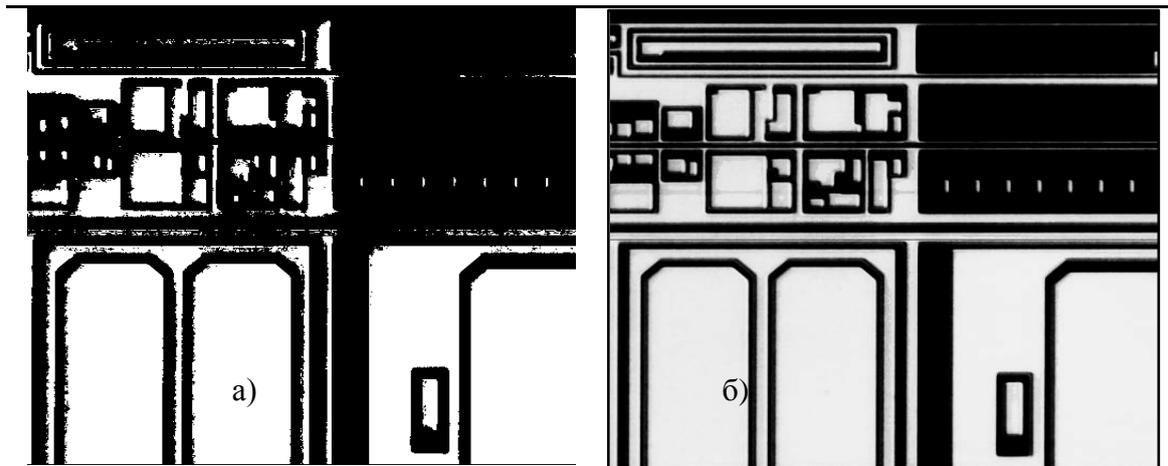


Рис 4. Изображение функции  $\chi_i(x, y)$ : а) - определенной по порогу на дисперсию; б) - определенной по функции  $c(x)$ .

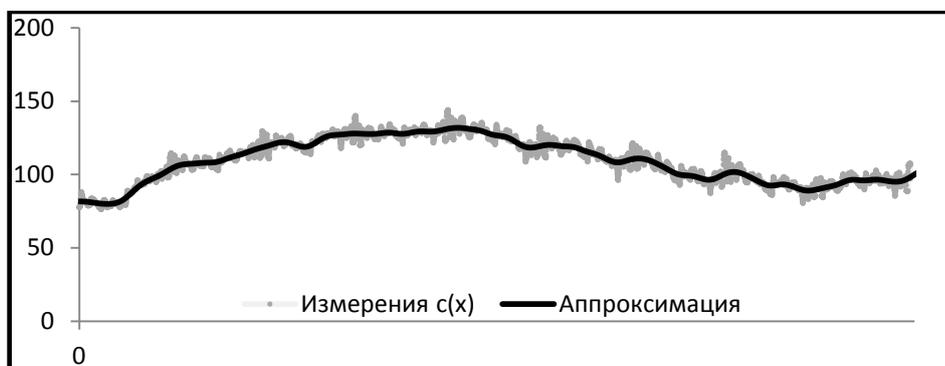


Рис 5. Аппроксимация функции  $c(x)$ .

Определив пороговое значение по минимуму функционала, можно определить функции  $\chi_i(x, y)$ , но лишь приблизительно из-за шумов и неточности модели (рис. 4а). Но этого достаточно для оценки функции  $c(x)$ , которая производится гладкой аппроксимацией по набору значений  $I(x, y)$  из точек  $(x, y): \chi_i(x, y) \neq 0$  (рис. 5).

Считая функцию  $c(x)$  известной, можно точнее приблизить  $\chi_i(x, y)$ , используя дисперсию отклонений  $\Delta_k = I(x + K \times k, y) - c(x + K \times k)$ ,  $k = 1..K$  и даже сопоставить каждой точке некоторый вероятностный вес, базируясь на данной величине (рис. 4, б). В итоге мы получим изображение (2.) с уменьшенной вариацией яркости, анализируя которое обычным методом поиска дефектов можно обнаружить

дефекты с существенно большей чувствительностью.

Для реализации описанного алгоритма было разработано экспериментальное программное обеспечение (ПО), которое в дальнейшем может быть встроено в штатное программное обеспечение установки ЭМ-6429Б. Результаты работы данного ПО представлены на рис. 6 а-в, где показаны изображения рис. 1 а-в после их обработки по описанному алгоритму. В результате дефект на первом кристалле (рис. 1, а) может быть легко обнаружен, т.к. разница яркости изображений в области дефекта после обработки составляет 28 о.е., тогда как разница в остальных областях не превышает 17 о.е. (рис. 6, г).

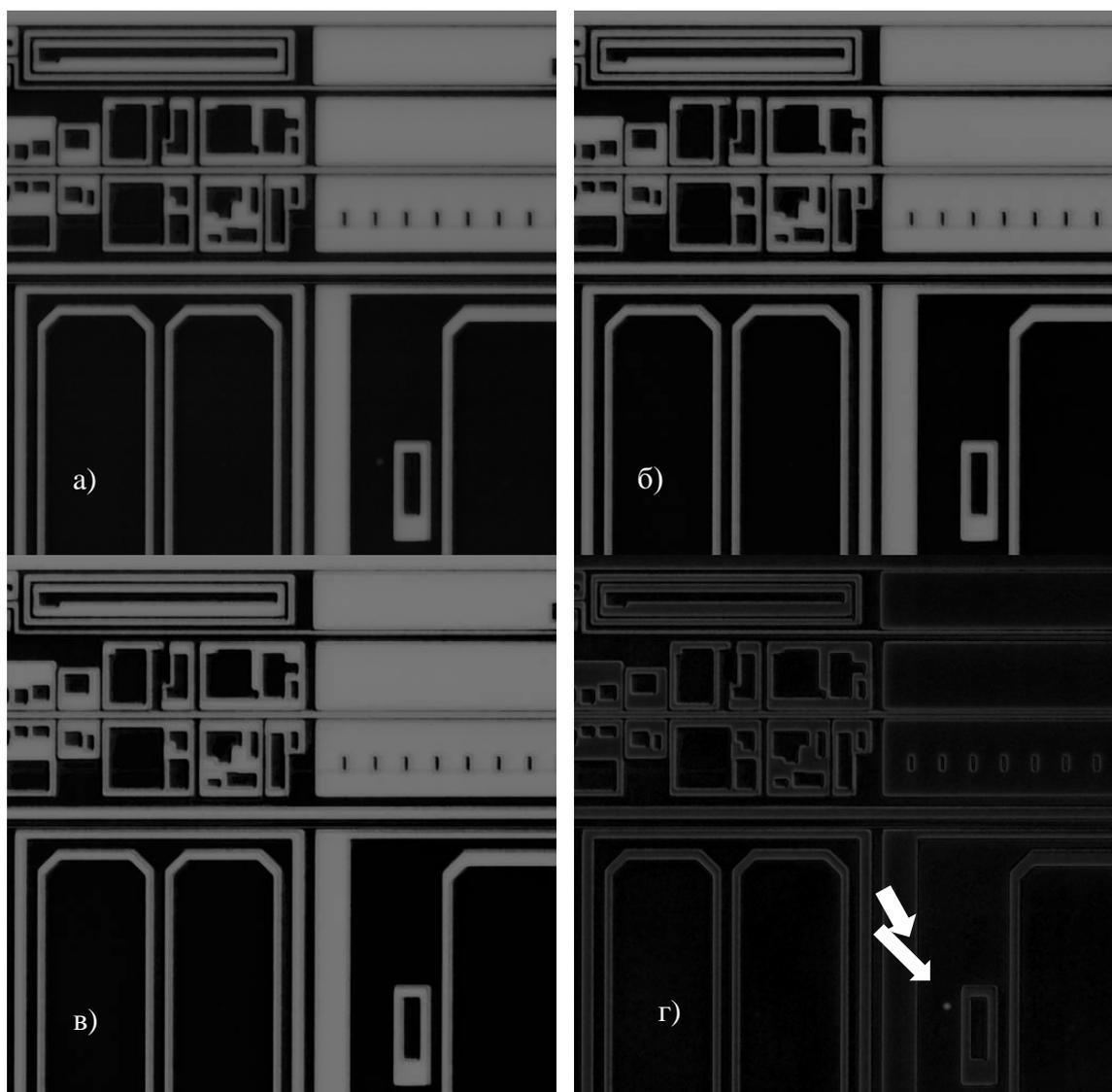


Рис 6. (а-в) - изображения участков пластины, после обработки алгоритмом по уменьшению вариаций, (г) - разница изображений (а) и (в), усиленная в 3 раза.

### 3. Выводы

В статье показана возможность существенно повысить чувствительность установки светлопольного контроля дефектности при наличии вариаций яркости изображения по пластине, обусловленных

присутствием тонких диэлектрических слоев различной толщины. Описан соответствующий метод обработки изображений и показан пример его реального применения для поиска дефектов с помощью установки ЭМ-6429Б.

## Improvement of brightfield inspection system sensitivity at the presence of image brightness variations

A.A. Zakharchenko, E.P. Mikhaltsov, M.V. Oreshkov, V.Yu. Troitskiy

**Abstract:** В статье описан метод повышения чувствительности установки светлопольного контроля дефектности на полупроводниковых пластинах с топологией, при наличии вариаций яркости изображения по пластине, обусловленных присутствием тонких диэлектрических слоев различной толщины. Показана реализация этого метода на основе установки ЭМ-6429Б в НИИСИ РАН.

The sensitivity improving method for brightfield defect inspection of semiconductor patterned wafers is developed. The method is suitable for wafers with brightness variations due to a presence of areas covered with transparent thin films with variable thickness along the wafer surface. An application of this method to EM-6429B equipment at SRISA RAS is demonstrated.

**Keywords:** defect inspection, improved sensitivity, design inspection, brightfield inspection, brightness variation

### Литература

1. А.А. Дьяченко, В.П. Рябухо Определение оптических толщин слоистых объектов по интерференционным цветам изображений в микроскопии белого света // Компьютерная оптика. – 2017. – Т. 41, № 5.
2. C. Perry-Sullivan, E. Le Roy, S. R. Lange, I. Malik, A. Yerabaka, A. Wilson, M. Shirey, B. Pinto, "Advantages of Broadband Illumination for Critical Defect Capture at the 65-nm Node and Below", Yield Management Solutions, Winter 2007
3. C. Perry-Sullivan, E. Le Roy, S. R. Lange, I. Malik, A. Yerabaka, A. Wilson, M. Shirey, B. Pinto Mark Shirey, Becky Pinto "Broadband Brightfield Inspection Enables Advanced Immersion Lithography Defect Detection" Yield Management Solutions, Spring 2007
4. А.А.Захарченко. Морфологические методы анализа многофокусных изображений. // Труды 12-й Международной конференции «Математические методы распознавания образов» ММРО-12, Москва 2005, ВЦ РАН, с. 335  
 . А.А.Захарченко, А.И.Чуличков. "Точность оценки и адекватность модели при минимаксном оценивании параметров формы сигнала" // Вестник Московского университета. Серия 3. Физика. Астрономия. 2006 №6 с. 11.

# Методы построения систем высокой готовности

А.Б. Бетелин<sup>1</sup>, А.А. Прилипко<sup>2</sup>, А.Г. Прилипко<sup>3</sup>, Г.А. Прилипко<sup>4</sup>, Д.В. Самборский<sup>5</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

E-mail's: <sup>1</sup> ab@niisi.msk.ru, <sup>2</sup> aaprilipko@niisi.msk.ru, <sup>3</sup> aleksey.prilipko@gmail.ru,

<sup>4</sup> prilipko@niisi.msk.ru, <sup>5</sup> dsambor@niisi.msk.ru

**Аннотация:** В статье рассмотрены основные понятия и методы создания и обеспечения высокой доступности и готовности сервисов, средства для создания и обслуживания кластеров высокой готовности, в том числе метрокластера и континентальных кластеров, обобщен накопленный опыт их эксплуатации, описан ряд наиболее часто возникающих при этом проблем и способы их решения.

**Ключевые слова:** распределенные системы, высокая доступность, кластера высокой готовности, метрокластер, континентальные кластеры, виртуальная серверная среда HPE, ServiceGuard.

## Введение

Для того, чтобы обеспечить необходимую доступность и надежность сервисов, нужно предпринять целый ряд действий, направленных на повышение надежности всех компонент предоставляющих их систем. Все это приводит к достаточно значительным затратам, в частности, из-за дублирования и повышения надежности сетевой инфраструктуры, систем обеспечения электропитания, применения более отказоустойчивых серверов.

При этом можно, например, определить допустимое время простоя в год. Так, стандартная надежность серверов (95%) предполагает до 18 дней простоев в год, причем серверы HPE (Hewlett Packard Enterprise) могут предоставить надежность не менее 98% (7.2 дня в год). Однако часто требуется высокая готовность (High Availability – HA), при которой должно обеспечиваться не более 12 часов незапланированного простоя (Media High Availability) в год. Однако это составляет всего лишь 99.86%, а высокий уровень HA - 99.95% (Top High Availability) предполагает простоя не более чем 4.3 часа в год. В случае же "пяти девяток" (99.999%) допускается всего 5 минут 15 секунд, что позволяет уже говорить об обеспечении отказоустойчивости (Fault Tolerant) сервера.

Повышение отказоустойчивости аппаратуры требует уже на этапах проектирования, разработки и изготовления предусмотреть средства для обнаружения отказов ее компонент и возможности восстановления их работоспособности после сбоев или дублирования. Все это может привести к достаточно значительному

увеличению ее стоимости и затрат времени. При этом необходимо учитывать, что отказы все равно возникают и важно заранее понимать, сколько времени может потребоваться на восстановление работоспособности системы. Возможно, потребуются замена каких-то компонент без остановки работы системы, например, «горячая замена» (hot swap) или их «горячее подключение» (hot plug).

При этом средства операционной системы (ОС) должны со своей стороны опознать произошедшее событие и выполнить необходимые действия, желательно без участия системного администратора. Для этого в современных ОС предусмотрен целый набор механизмов. Так, поддержка мультипассинга (multipathing) позволяет обеспечить автоматическое переключение передачи данных по другим каналам ввода/вывода системам хранения данных, агрегация портов (bonding/teaming) позволяет обнаружить отказы сетевых плат и обеспечить непрерывность работы в сети. Кроме того, в современных ОС динамически загружаются новые модули ядра и драйверы для вновь обнаруженных устройств. При отказе и замене диска с помощью программного или аппаратного RAID (Redundant Array of Independent Disks - избыточный массив независимых дисков) автоматически могут быть восстановлены данные. При этом важно сразу же после отказа диска произвести необходимые действия для восстановления полной работоспособности RAID. Интересно отметить, что сейчас вместо RAID-5 часто используется RAID-6, который содержит две контрольные суммы и позволяет не потерять данные в том случае, если в процессе перестройки данных произойдет выход из строя еще одного диска и предоставляет системному администратору

некоторый дополнительный запас времени.

Есть дорогие и надежные решения с полным дублированием большинства компонент сервера, в том числе и процессоров. Так, уже в 1976 году компанией Tandem Computer Inc. была предложена линейка серверов NonStop с использованием системной коммуникационной сети ServerNet под управлением специализированной операционной системы и набора программного обеспечения NonStop. В частности, в этих серверах три процессора выполняют одинаковые потоки команд и рассматриваются как один процессор, поэтому отказ любого из них не является критичным и заметным для приложений. Синхронизация происходит при обработке внешних прерываний и во время обращения к глобальной памяти, что позволяет обнаружить отказ процессора. В настоящее время компания стала одним из подразделений HPE, а линейка HPE NonStop [1] включает серверы на базе процессоров Intel Itanium и Xeon. Кроме того, на базе платформы HPE Virtualized NonStop в частном облаке предприятия можно развернуть специализированное ПО NonStop. Обычно такие решения используются там, где необходима абсолютная надежность, полная отказоустойчивость и доступность сервиса.

## Кластеры компьютеров

Более экономичным и достаточно эффективным вариантом для обеспечения высокой степени доступности сервиса является создание кластера высокой готовности, который представляет собой набор совместно работающих узлов (серверов) для повышения доступности приложений. В этом случае при возникновении SPOF (single point of failure) — единой точки отказа, например, при остановке сервера, происходит автоматический перезапуск приложений на других узлах кластера. При этом некоторое время сервис может быть недоступен. Исключением являются приложения, специально предназначенные для работы на нескольких серверах одновременно. Так, для СУБД Oracle еще в 2001 году появилось программное обеспечение (ПО) Oracle Real Application Clusters (RAC) для кластеризации и повышения доступности базы данных. По мере развития она стало называться Oracle Clusterware. Создание такого ПО позволяет уменьшить время недоступности сервиса, поскольку не требуется заранее непредсказуемое время на восстановление базы данных при перезапуске

СУБД на другом узле, а нагрузка просто перераспределяется между оставшимися узлами. Однако это дополнительно требует согласованного совместного доступа к файлам БД, а также постоянного согласования находящейся в оперативной памяти разных узлов SGA (System Global Area) в запущенных на разных узлах «экземпляров» (Instances) СУБД Oracle. Каждый из них представляет собой независимый набор процессов и сегмент разделяемой оперативной памяти.

Использование кластеров высокой готовности дополнительно позволяет уменьшить продолжительность запланированных простоев, связанных с обслуживанием серверов, установкой и обновлением ОС или приложений. Так, при аппаратном обновлении сервера (upgrade) приложения с него предварительно могут быть практически без остановки перемещены на другие узлы кластера.

## HPE ServiceGuard

В данной статье мы в первую очередь будем рассматривать решения компании HPE для создания кластеров высокой готовности, поскольку одним из авторов статьи накоплен более чем пятнадцатилетний опыт работы с ними.

Первая версия программного продукта MC/ServiceGuard (ServiceGuard) появилась еще в 1990 году для HP-UX и была на тот момент единственным кластером высокой готовности для операционных систем UNIX. Одним из важных его свойств является то, что он представляет собой обычное приложение в ОС и под его управлением могут находиться практически любые приложения. Одновременно в ряде случаев для его работы были добавлены специальные возможности в ОС, так, например, в операционной системе HP-UX 10.20 не было штатной возможности активации нескольких IP адресов на одной сетевой карте, но средства, предоставляемые в рамках продукта ServiceGuard, позволяли ее обеспечить.

## Создание кластера

При планировании развертывания кластера необходимо определить, какие именно приложения и на каких узлах будут запускаться под его управлением и какая для этого необходима инфраструктура. После подготовки ОС, включающую в себя установку

необходимого программного обеспечения, настройку сети, обеспечение доступа к общим СХД, на каждом из серверов кластера создается специальный файл `smcnodelist`, содержащий перечень узлов кластера.

После чего вызывается специальная программа для создания конфигурационного файла кластера.

В нем указывается имя кластера, информация о том, какие именно сетевые карты будут использоваться для отправки служебной информации, в частности сигналов сердцебиения (`heartbeat`), какие задержки (`timeout`) допустимы, какой из механизмов для определения кворума будет задействован при необходимости, а также информация, описывающая узлы кластера.

После проверки и применения данной конфигурации создается бинарный файл `smclconfig`, актуальное состояние которого копируется на все доступные узлы кластера.

Он используется при работе кластера и в случае необходимости по нему можно получить текстовые конфигурационные файлы, хотя полезно (*best practices*) регулярно копировать их на другие узлы.

Для управления приложениями в ServiceGuard введено понятие пакета (`package`), которое позволяет рассматривать как единое целое приложение и все необходимые для его работы ресурсы, в число которых входят IP адреса для доступа к нему, место для хранения данных (устройства, файловые системы, группы томов LVM или дисковые группы VxVM), сервисные процессы и т.п.

При запуске пакета на одном или нескольких узлах кластера сначала активируются все необходимые для его работы ресурсы, после чего запускается приложение с помощью контрольного скрипта на языке shell. Важно отметить, что если при старте пакета возникает проблема, например, невозможна активация группы томов, то происходит его остановка и освобождение ресурсов, после чего осуществляется попытка запуска приложения на другом узле кластера.

При запуске кластера входящие в него узлы, а вернее демоны кластера (`smclsd`), обмениваются между собой служебной информацией и определяют, кто из них будет координатором кластера и пакетов.

В дальнейшем они постоянно проверяют, приходят ли специальные `heartbeat` пакеты от других узлов, наличие которых означает, что «соседи живы».

В случае остановки или на время перезагрузки какого-то узла он помечается как недоступный (`down`), а его пакеты будут

запущены, если это окажется возможным, на других узлах.

## Кворум

Более сложная ситуация возникает в том случае, если кластер распадается на части, например, при возникновении проблем с сетями, предназначенными для отправки служебной информации, или с истечением времени ожидания получения `heartbeat` пакетов. При этом каждая из частей попытается самостоятельно сформировать свой кластер и запустить на нем недостающие пакеты. Если в какой-то из частей оказалось менее половины узлов, то считается, что все серверы должны быть немедленно остановлены, для чего демон кластера на каждом из этих узлов посылает сигнал своей операционной системе на ее принудительную быструю остановку. В результате ресурсы, например, активированные в эксклюзивном режиме группы томов LVM, освобождаются и могут быть использованы при запуске пакетов на других узлах. Если же кластер распался на две части с одинаковым числом узлов, происходит обращение к кворум-серверу или кворум-диску. В результате этого обращения та часть, которая первой получит кворум и сформирует новый кластер, причем узлы другой части будут принудительно остановлены. Это позволяет избежать ситуации так называемого `split-brain`, когда несколько узлов пытаются несогласованно работать с одними и теми же данными. Отметим, что если по каким-то причинам кворум-сервер в этот момент недоступен, то будут остановлены все узлы кластера.

Поэтому одним из ключевых моментов при создании кластера является обеспечение надежности сетевой инфраструктуры, используемой для обмена служебной информацией, в том числе так называемыми `heartbeat` пакетами, а также доступности по сети кворум-сервера, который представляет собой отдельный сервер с постоянно запущенной на нем программой `qs`, предназначенной для обслуживания запросов от узлов кластера (кластеров). Если узлы кластера находятся в нескольких центрах данных (ЦОД), каждый со своей системой хранения данных (СХД), то желательно, чтобы кворум-сервер находился еще на одной дополнительной площадке, что позволяет в случае выхода из строя какого-нибудь из центров обработки данных сформировать кластер из узлов, находящегося в другом ЦОД. Для повышения надежности программа `qs` может находиться в одном из пакетов другого

кластера. Отметим, что использование в качестве кворума диска или LUN дискового массива ограничивает число допустимых узлов в кластере четырьмя, а при отказе сети в качестве узлов для формирования кластера случайно могут быть выбраны те, у которых нет доступа к сети.

В результате переконфигурирования кластера число работающих узлов в нем уменьшится и возникнет задача перевыборов координаторов кластера и пакетов и перезапуска пакетов, работавших на остановленных узлах.

## Пакеты

При описании пакета в ранних версиях ServiceGuard создавался его конфигурационный файл, а также контрольный скрипт на языке shell, который вызывался как при старте, так и при остановке пакета. Однако, начиная с версии 11.18, дополнительно появились так называемые модульные пакеты с единым конфигурационным файлом для пакета, в котором задаются все необходимые параметры для работы пакета. Данные параметры хранятся в конфигурационном файле кластера и используются при запуске соответствующих скриптов на языке shell, которые при этом остаются неизменными и одинаковыми на всех узлах. По существу, переменные, которые ранее находились внутри программы на языке shell, стали параметрами пакета и хранятся в общем бинарном файле кластера `smclconfig`. Такой подход существенно упростил создание инструментов для конфигурирования и управления кластерами и пакетами, в том числе и через web интерфейс. Кроме того, улучшился контроль за параметрами, а также появились дополнительные возможности изменения параметров без остановки пакета.

Первоначально пакет в ServiceGuard мог запускаться только на одном узле, а в случае необходимости происходило его перемещение на другой узел (`failover`). Все доступные для пакета узлы перечисляются в его конфигурации, причем первым указывается основной (`primary`) узел, на котором приложение запускается при старте кластера, а остальные считаются запасными (`adaptive`) и в случае `failover` выбираются в том порядке, в котором они указаны.

Помимо такого режима по умолчанию (`configured_node`) в качестве политики `failover_policy` может быть указан `min_package_node`, при котором пакет запускается там, где меньше всего пакетов. Это позволяет, в частности, реализовать режим

`rotation standby`, когда есть дополнительный узел, который находится в режиме ожидания, и при отказе любого из узлов запускает у себя его пакеты.

Кроме того, начиная с версии 11.18, в ServiceGuard появилась возможность создания пакетов, которые запускаются одновременно на нескольких (`multi_node`) или на всех узлах (`system_multi_node`) кластера. В первую очередь они были предназначены для обеспечения функционирования Oracle RAC и кластерной файловой системы CFS.

## Сервисные процессы (сервисы)

Интересной особенностью ServiceGuard является введение понятия сервиса, представляющего собой процесс (часто называемый сервисным), наличие которого необходимо для продолжения работы приложения на данном узле. У каждого пакета их может быть до 30. Они запускаются при старте пакета перед запуском приложения, демон кластера `smcld` запоминает уникальные идентификаторы этих процессов (`pid` в операционной системе UNIX) и с частотой раз в 10 секунд проверяет их наличие. При отсутствии какого-нибудь сервиса демон кластера `smcld` пытается его перезапустить. При конфигурации сервиса можно ограничить число таких попыток или даже запретить его перезапуск. В случае неудачи возникает отказ сервиса и пакет должен запускаться на другом узле, причем при необходимости может быть принудительно остановлен тот узел, на котором не удалось продолжить выполнение приложения.

Отметим, что поскольку доступность сервиса отслеживается по его уникальному идентификатору (`pid` процесса), то при нормальной ситуации этот `pid` не должен меняться или завершаться. Поэтому часто используется такой подход: в качестве сервиса запускается специальный скрипт для мониторинга, который следит за приложением, например, за процессами СУБД, в случае возникновения каких-то проблем пытается их исправить, а в случае неудачи завершается. Поскольку в конфигурации этого сервиса указано, что он не должен запускаться заново, то считается, что возник отказ и приложение должно быть запущено на другом узле. В результате демон кластера следит только за этим сервисом, а не за всеми процессами, которые необходимы для работы приложения. Разработчики сложных программных продуктов, в том числе СУБД, совместно с HPE

разрабатывают и предоставляют свои собственные скрипты и модули для ServiceGuard, которые позволяют корректно стартовать/останавливать их приложения, а также осуществлять непрерывный мониторинг за правильностью их функционирования. Это позволяет более полно учесть все особенности поведения приложения и накопленный опыт его эксплуатации.

## Мониторинг ресурсов

Еще одной особенностью кластера является его интеграция с EMS (Event Monitoring Service), который позволяет осуществлять мониторинг ресурсов, проверять их статус и посылать извещение о наступлении некоторых сконфигурированных событий.

При конфигурации пакета можно указать, что необходимо отслеживать, с каким интервалом и что делать при невыполнении заданных условий. Так, при отказе конкретной сетевой карты или недостатке свободного места в некоторой файловой системе пакет может уйти на другой узел или даже «громко хлопнуть дверью» - принудительно остановить узел, что приведет и к перемещению других пакетов.

## Выбор узла при failover

При выборе узла, на котором будет запущен пакет, дополнительно проверяется, разрешен ли ему запуск на нем, так, если он на данном узле уже был, но из-за отказа сервиса ему пришлось уйти, в результате чего был выставлен флаг, что данный узел для него недоступен.

Поэтому одним из важных аспектов при работе кластера является отслеживание того, является ли пакет перемещаемым (установлен ли флаг `auto_run`) и есть ли для него доступные узлы, кроме того, на котором он запущен в данный момент.

При запуске пакетов дополнительно учитывается целый ряд факторов, в том числе емкости узлов и вес пакетов, их приоритеты, порядок их запуска, зависимости между ними, в частности, должны ли они запускаться на одном или разных узлах.

Поэтому, если есть такая возможность, желательно перед запуском или перемещением пакета запустить соответствующие команды в режиме тестирования (дополнительный ключ `-t` у команды) или проверить, что произойдет при изменении состояния кластера с помощью

специальной команды `smeval`.

Для этого текущее состояние кластера сохраняется в текстовом файле (`smviewcl -f line`), в него вносятся необходимые корректировки, выполняется данная команда и проводится анализ ситуации.

## Обнаружение неисправностей и обслуживание кластера

Важным фактором при работе с кластером является наличие, помимо обычных, еще и специальных инструментов для мониторинга, тестирования, идентификации и исправления проблем. Для проверки того, в порядке ли инфраструктура, можно использовать команды, используемые при создании кластера, например, `smquerycl` и `smcheckconf`, которые позволяют понять, не возникли ли какие-нибудь новые проблемы. Команды `smviewconf` и `smgetconf` могут получить текстовую конфигурацию по бинарному файлу и работающему кластеру. Команда `smcompare` позволяет сравнивать файлы из заданного списка на разных узлах кластера. С помощью `smgetpkgenv` можно получать атрибуты пакета в виде переменных shell, которые могут применяться при автоматизации мониторинга кластера.

При отладке запуска модульных пакетов можно указать, какие именно модули пакетов надо выполнить. Кроме того, пакет может быть переведен на некоторое время в специальный режим обслуживания (*maintenance mode*), в котором некоторые из событий в кластере им игнорируются, перемещение пакета автоматически не происходит, при перезапуске узла или кластера он не стартует и т.п.

В последних версиях ServiceGuard появилась возможность остановить функционирование кластера на узле или даже на всех узлах без остановки запущенных приложений. Такие пакеты при последующем старте узла и кластера будут автоматически переданы под управление кластера. Данный подход позволяет производить обслуживание серверов или кластера без остановки приложений. Более того, изменение большей части параметров как пакетов, так и кластера, например, замена кворум-механизма, в настоящее время может осуществляться без их остановки. Информация об основных событиях в кластере попадает в общий файл протокола ОС. Кроме того, для каждого пакета есть свой файл протокола, имя которого указано в конфигурации пакета, что позволяет, в частности, нескольким пакетам писать

протокол в единый файл. При этом в конфигурации пакета можно указать свой собственный уровень подробности сообщений, который может быть изменен также и с помощью дополнительного ключа при запуске ряда команд.

## Метрокластер и континентальные кластеры

Помимо продуктов, обеспечивающих работу локальных кластеров, в семейство ServiceGuard входят также средства для создания метрокластеров и континентальных кластеров.

Метрокластер обычно размещается в нескольких ЦОД, объединенных между собой с помощью нескольких оптоволоконных кабелей и находящихся друг от друга на расстоянии до 100 и даже до 300 километров, что позволяет обеспечить устойчивость приложений не только возникновению одиночного отказа (SPOF), но и при отказе центра данных целиком. Особенностью данного решения является наличие в каждом ЦОД собственной СХД и аппаратных средств синхронизации между ними в режиме «активный/пассивный» с возможностью автоматического переключения режима доступа к LUN на дисковых массивах при переходе приложения с одного узла на другой. Кроме того, предусматривается наличие отдельной площадки, на которой расположены средства, позволяющие при необходимости получить кворум (арбитраторы или кворум-серверы).

В ряде случаев возможности, предоставляемые метрокластером, можно получить на основе обычного кластера, используя программные средства для зеркалирования данных на LUN в СХД, располагающихся в разных ЦОД.

Континентальные кластеры (HPE Continental Clusters) представляют собой несколько (не менее двух) полноценных кластеров, соединенных между собой по обычной сети. На каждом из них выполняется свой набор пакетов. Для них он является основным и называется Primary Cluster, однако при катастрофе и отказе кластера целиком они могут быть запущены на заранее подготовленном для них резервном кластере (Recovery Cluster), на котором помимо своих пакетов, запущены пакет мониторинга состояния основного кластера, пакеты для приема данных приложений, которые посылаются специальным пакетом основного кластера, а также сконфигурированы пакеты, необходимые для запуска приложений,

пришедших с основного кластера. Можно организовать взаимное перемещение пакетов, когда для ряда пакетов основным является один кластер, а для других – другой кластер. На узлах кластеров помимо ServiceGuard установлено дополнительное программное обеспечение (HPE Continental Clusters), позволяющее обеспечить взаимодействие между кластерами. Важно отметить, что переход пакетов осуществляется в полуавтоматическом режиме. Поэтому часто в качестве основного кластера используется метрокластер, и только в случае выхода из строя всех его ЦОД выполняется необходимый комплекс мероприятий для запуска приложений на резервном кластере. Сразу же после получения сообщений об отказе основного кластера системный администратор резервного кластера связывается с администраторами основного кластера и при подтверждении запускает команду для перехода пакетов (*cmrecovercl*). Важно понимать, что вообще говоря, какие-то потери данных при этом все-таки могут произойти, но восстановление доступности приложений будет достаточно быстрым и полноценным.

В основном файле конфигурации для континентальных кластеров описывается, какие кластеры и узлы в него входят, каким образом осуществляется мониторинг, синхронизация конфигурации кластера и куда посылаются сообщения. Кроме этого, в нем описываются группы восстановления пакетов (recovery groups), в каждую из которых входит основной (primary) пакет, пакет для восстановления (recovery) и, при необходимости, пакеты для отправки (sender) и приема (receive) данных, и возможно, репетиционный (rehearsal) пакет. Перед именами пакетов дополнительно указывается имя того кластера, на котором он первоначально запускается.

Одной из важных особенностей HPE Continental Clusters является возможность тестирования ситуации восстановления при работающем основном кластере, называемой репетицией или «примеркой одежды» (Dress Rehearsals). Это позволяет получить необходимый опыт, найти слабые места, убедиться, что все прошло в «штатном» режиме, наладить взаимодействие между группами администраторов, отвечающих за разные кластеры, что особенно важно при расположении кластеров в разных странах. При этом главное - не помешать работе основного кластера. Для чего предусмотрена возможность перевода группы восстановления пакетов в режим обслуживания. В этом случае приостанавливается передача данных от

основного сервера к резервному и не может быть выполнен перевод пакета, но запускается репетиционный пакет, после чего изучаются возникшие проблемы.

Отметим, что одной из проблем при развертывании континентальных кластеров и проведения репетиций, является обеспечение доступа к IP адресам пакетов, а точнее к их именам. При этом могут использоваться как DNS, LDAP механизмы, так и динамически маршрутизируемые скрытые подсети (dynamically routed hidden subnet) с указанием большого числа переходов (hop) до них или механизма изменения IP адресов (destination address mangling).

## Виртуальная Серверная Среда

Еще одной важной особенностью ServiceGuard является его тесная интеграция с Виртуальной Серверной Средой HP (VSE HP) [2]. Так, виртуальные машины HP (Integrity VM) могут использоваться как узлы кластера, так и внутри пакетов, для создания которых предусмотрен специальный инструментарий (toolkit), в том числе введено понятие соответствующих ресурсов для менеджера виртуальных машин. Кроме того, при необходимости с помощью специальных команд можно перекидывать процессоры между vPag (виртуальными разделами) или лицензии Instant Capacity для nPAR (аппаратные разделы) в том случае, если какой-нибудь из узлов вышел из строя. Возможно

также и автоматическое переконфигурирование ресурсов благодаря соответствующим настройкам WLM (Workload Manager), специального приложения, которое позволяет следить за загрузкой системы и предоставлять ресурсы, чтобы обеспечить необходимое время отклика.

Конфигурация и мониторинг кластеров может осуществляться в рамках web интерфейса SMH (System Management HomePage) и HPE SIM (Systems Insight Manager).

## Заключение

В статье были рассмотрены основные понятия и методы создания и обеспечения высокой доступности и готовности сервисов, средств для создания и обслуживания кластеров высокой готовности, обобщен накопленный опыт их эксплуатации, описан ряд наиболее часто возникающих при этом проблем и способы их решения.

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (выполнение фундаментальных научных исследований ГП 14) по теме № 0065-2018-0004 "Развитие методов математического моделирования распределенных систем и соответствующих методов вычисления." (№ АААА-А18-118041290146-7).

# The principles of design for highly available systems

A.B. Betelin, A.A. Prilipko, A.A. Prilipko, G.A. Prilipko, D.V. Samborskiy

**Abstract:** This article reviews the basic concepts and methods for creating and ensuring high availability services. It also describes tools for building and maintaining high availability clusters, including metrocluster and continental clusters. We analyze the most frequently encountered problems and present trouble-shooting approaches based on gained experience.

**Keywords:** distributed systems, high availability, high availability clusters, metrocluster, continental clusters, Virtual Server Environment HPE, ServiceGuard.

## Литература

1. HPE Integrity NonStop. URL: <https://www.hpe.com/ru/ru/servers/nonstop.html>. Дата обращения: 07.11.2018
2. Дэн Эрингтон, Брайен Джаккуот. Виртуальная Серверная Среда HP. Сделайте Адаптивную Инфраструктуру реальностью в Вашем центре обработки данных. Перевод с английского Г.А.Прилипка. ISBN 978-6-9556-0103-8, 502с.

# Оптимизация архивирования виртуальных операционных систем в среде GNU Linux/QEMU/KVM/Libvirt

А.Б. Бетелин<sup>1</sup>, А.А. Прилипко<sup>2</sup>, Г.А. Прилипко<sup>3</sup>, С.Г. Романюк<sup>4</sup>,  
Д.В. Самборский<sup>5</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

E-mail's: <sup>1</sup> [ab@niisi.msk.ru](mailto:ab@niisi.msk.ru), <sup>2</sup> [aaprilipko@niisi.msk.ru](mailto:aaprilipko@niisi.msk.ru), <sup>3</sup> [prilipko@niisi.msk.ru](mailto:prilipko@niisi.msk.ru),

<sup>4</sup> [sgrom@niisi.ras.ru](mailto:sgrom@niisi.ras.ru), <sup>5</sup> [dsambor@niisi.msk.ru](mailto:dsambor@niisi.msk.ru)

**Аннотация:** Среда виртуализации QEMU/KVM/Libvirt в операционной системе GNU Linux позволяет архивировать состояние виртуальных операционных систем (ОС), не прерывая их исполнения. В данной статье предложены методы создания и хранения инкрементальных архивов состояний виртуальных ОС, которые позволяют экономить объем занимаемых данных, повысить надежность архивов, и увеличить скорость архивирования. Анализ двухлетнего опыта эксплуатации архива виртуальных ОС показал, что при правильном использовании инструменты QEMU/KVM/Libvirt не уступают по основным критериям современным коммерческим системам виртуализации.

**Ключевые слова:** Virtualization; Linux; KVM; QEMU; Libvirt; copy-on-write; incremental backup; userfaultfd.

## Введение

Под технологией виртуализации операционных систем в настоящее время понимается совокупность программных и аппаратных средств, позволяющих создать среду исполнения для некоторой операционной системы. Эта среда должна быть неотличима от той, которая обычно требуется для исполнения этой ОС. Операционная система, работающая внутри среды виртуализации, называется "гостевой" или "виртуальной". Операционная система, образующая систему виртуализации, называется гипервизором или "хост-системой". Виртуализация не является новой технологией, в виде полной программной эмуляции работы одного компьютера средствами более производительного компьютера виртуализация применялась уже десятилетия назад, она позволяла продлить эксплуатацию программных комплексов в сфере бизнеса и телекоммуникаций. Тем не менее, подобная эмуляция работала существенно медленнее, чем выполнение кода в хост-системе.

Современные технологии виртуализации позволяют почти полностью избежать потерь в скорости выполнения процессорного кода, используя динамическую бинарную компиляцию ("just-

in-time", JIT-компиляция), либо используя тот факт, что процессорная архитектура гостевой системы поддерживается процессорами гипервизора. Примерами последнего случая служит технология паравиртуализации в среде Xen [9] и использование аппаратной поддержки виртуализации гипервизором Kernel-based Virtual Machine, KVM [1]. В обеих этих средах программный код архитектур x86 или amd64 может работать на гипервизоре с архитектурой amd64 со скоростью, близкой к номинальной на используемом процессоре [10]. Для обеспечения эффективной работы систем ввода-вывода данных современные среды виртуализации предлагают специальные драйверы виртуальных устройств, использующие технику паравиртуализации. В этом случае драйвер, например, дискового накопителя не представляет известный интерфейс шины ввода-вывода (SCSI, SATA, IDE), а напрямую указывает буфер данных и команду записи или чтения библиотеке гипервизора, которая выполняет чтение или запись без лишних накладных расходов.

Достаточно высокая эффективность исполнения виртуальных операционных систем делает технологию виртуализации привлекательной для решения целого комплекса задач эксплуатации вычислительных систем. Так, современные

технологии виртуализации операционных систем позволяют:

- оптимальнее использовать вычислительные ресурсы, консолидировать аппаратные ресурсы обработки, хранения, и передачи информации для снижения общей стоимости эксплуатации вычислительных систем;
- обновлять вычислительные узлы без изменения логической структуры информационных систем;
- продолжать использовать устаревающие операционные системы, выполняющие критически важные приложения;
- выполнять резервное копирование операционных систем (в том числе с сохранением текущего состояния прикладных систем);
- разделять сложные прикладные системы на более простые изолированные контейнеры;
- обеспечивать отказоустойчивую работу сервисов с горизонтальным масштабированием и балансировкой нагрузки;
- создавать виртуальные системы для однократных или тестовых задач.

Несмотря на такое разнообразие задач, решаемых виртуализацией, виртуальные ОС делятся на две категории: с хранимым внутренним состоянием (stateful) и без хранимого внутреннего состояния (stateless). Например, к последней категории относятся системы серверов приложений, а также веб- и файл-серверы, раздающие данные, хранимые во внешних базах данных или дисковых массивах. Функция таких систем состоит в обработке запросов пользователей и представлении ответов в нужном формате. Экземпляры таких ОС могут быть взаимозаменяемы, и запускаются в том количестве, которое необходимо для обработки потока запросов.

Виртуальные ОС с хранимым состоянием более сложны в эксплуатации, поэтому их использования стараются избегать при проектировании программных комплексов, сохраняя данные в системах, специально для этого спроектированных – распределенных файловых системах и базах данных. Но ОС с хранимым состоянием могут быть унаследованными, призваны выполнять важные функции и поэтому требуют сопровождения и развития. В этом случае приходится эксплуатировать виртуальные ОС, требующие решения всех

административных задач, включая их своевременное резервное копирование.

## Способы архивирования

Резервное копирование ОС или прикладной вычислительной системы призвано:

- дать возможность восстановить архивные файлы или данные по состоянию на некоторые моменты времени (в идеале – на произвольный момент времени в прошлом);
- минимизировать вмешательство в работу архивируемой системы (время недоступности или периода сниженной производительности в процессе создания архива);
- минимизировать количество дополнительных ресурсов, необходимых для резервного копирования (дискового пространства, каналов передачи информации, вычислительных ресурсов).

Одновременно выполнить все эти требования невозможно, поскольку они противоречат друг другу. Тем не менее, все современные среды виртуализации имеют функцию безостановочного архивирования состояния ОС, а также возможности хранения цепочек архивов виртуальных дисков (так называемые "инкрементальные" архивы). Возможные стратегии резервного копирования виртуальных ОС различаются способами создания архивов и способами их последующего хранения. Архивировать состояния виртуальной ОС можно следующими способами (см. Таблицу 1):

- Полное выключение ("offline"), после которого сохраняются копии виртуальных дисков выключенной виртуальной ОС.
- Копирование дисков приостановленной ОС ("freeze") – при этом сохраняются копии виртуальных дисков приостановленной ОС после сброса буферов ввода-вывода.
- Создание точного "снимка" ("memory snapshot") – в этом случае сохраняются копии виртуальных дисков приостановленной ОС, а также снимок всей оперативной памяти виртуальной системы и состояния регистров и буферов виртуальных устройств ввода-вывода.

Для метода "offline" архивирование требует завершения всех прикладных программ этой ОС и выключения самой системы. После архивирования ОС и прикладные программы запускаются в порядке, обратном к порядку выключения.

"Freeze" метод полагается на то, что перед приостановкой виртуальной ОС на виртуальные диски сбрасываются все несохраненные данные из буферов прикладных систем, страниц файловых кэшей и системных буферов ввода-вывода.

Таблица 1. Методы архивирования виртуальных ОС.

Метод архивирования	Offline	Freeze	Memory snapshot
Необходимость остановки гостевой ОС	+	-	-
Поддержка со стороны гостевой ОС и прикладных программ (guest agent, user freeze scripts)	-	+	-
Время недоступности	большое	малое	среднее

Например, для сервера базы данных, отвечающего требованиям ACID, это действие должно приостановить обработку запросов и завершить выполнение всех текущих транзакций, после чего достаточно выполнить запись модифицированных страниц файлового кэша и дисковых буферов. За эту функцию отвечает специальная служба поддержки виртуализации (guest agent) и она может быть дополнена специальными инструкциями для выполняемых прикладных систем. Не все прикладные системы позволяют правильно приостановить свою работу с сохранением всех данных на диск, поэтому данный метод архивирования не является универсальным. Тем не менее, это наиболее быстрый способ архивирования виртуальных ОС (особенно в сочетании с приемами создания моментального снимка виртуальных дисков, см. ниже).

Метод "memory snapshot" имеет преимущество в том, что для архивирования не требуется прерывать работу прикладных программ, и невелико время приостановки ОС (оно зависит от размера оперативной памяти и способа копирования виртуальных дисков, см. ниже). Это единственно возможный способ в ситуации, когда нельзя выключать архивируемую ОС и либо для нее нет агента виртуализации, либо нет

возможности выполнить сброс буферов прикладных систем так, чтобы не осталось несохраненных на диск транзакций. Архивы виртуальных дисков обычно занимают значительный объем, поэтому имеет смысл оптимизировать их хранение тем или иным способом. Информация, содержащаяся в архивах виртуальных дисков, обычно избыточна по двум причинам: (1) данные внутри виртуального диска имеют низкое информационное содержание (низкую энтропию) либо виртуальный диск не полностью заполнен данными; (2) последовательные архивные копии виртуальных дисков часто имеют лишь небольшие отличия.

Экономия объема архивов за счет компрессии данных возможна, но имеет ряд недостатков и усложняет доступ к архивам. Кроме того, в случае не полностью заполненного диска последовательность неинициализированных данных по умолчанию считается нулевыми байтами и не хранится в большинстве форматов виртуальных дисков.

Напротив, избыточность, заключенная в последовательности состояний дисков, может быть использована для значительной экономии дискового пространства, без заметного усложнения процедур создания и использования архивов. Для этого либо вся файловая система архива представляется в виде последовательности снимков, либо изменения данных, сделанные в каждом виртуальном диске, хранятся в отдельных файлах.

В первом варианте полные архивы дисков сохраняются в файлах под одними и теми же именами, а затем фиксируются в виде снимков состояния всей файловой системы архива. В среде ОС Linux это можно сделать в файловых системах ZFS, BTRFS, или средствами системы управления логическими томами LVM thin provisioning. При этом важно обеспечить обновление сохраненных архивов без их копирования или сдвига ("in-place" обновление), иначе не получится достичь экономии. Это требование нелегко выполнить, особенно если используется сложный формат виртуального диска (VMDK или QCOW2). Примером "правильного" обновления служит копирование диска в формате RAW командой `rsync` с опцией `'-inplace'`, тогда при очередном копировании в файле обновляются только измененные блоки. Преимущество данного подхода состоит в простоте использования архива, поскольку снимки состояния создаются сразу для всего хранилища архивов.

Недостатками этого подхода являются трудность планирования расширения объема хранилища и невозможность использования разной частоты архивирования и срока хранения архивов для разных виртуальных ОС. Кроме того, скорость работы подсистемы LVM снижается с увеличением количества снимков, а файловые системы ZFS и BTRFS в системе Linux не достигли достаточной степени надежности (отметим, что компания RedHat сворачивает поддержку файловой системы BTRFS).

Во втором варианте используются снимки виртуальных дисков, которые представляют собой отдельные файлы, содержащие карты измененных блоков и их содержимое, соответствующие изменениям, сделанным в этом снимке по сравнению с базовым для данного снимка образом. Таким образом, снимки образуют связанные цепочки файлов, отражающие серию сохраненных состояний виртуального диска. Первый файл в цепочке файлов может быть файлом виртуального диска текущего состояния виртуальной ОС, тогда его содержимое непрерывно дополняется изменяемыми данными в результате работы этой ОС (так называемая техника "дубликации при записи", "copy-on-write"). Функции создания последовательности снимков виртуальных ОС есть во всех современных средах виртуализации. Далее предполагается, что используется система виртуализации с открытым исходным кодом QEMU/KVM, дополненная инструментами Libvirt в операционной системе Linux [1,2,3].

Формат виртуальных дисков QCOW2, используемый QEMU, позволяет создавать внутренние и внешние снимки. С точки зрения системы виртуализации эти типы снимков почти не отличаются. Внешние снимки – это отдельные файлы QCOW2, которые имеют ссылку на другой файл виртуального диска, как на базовый образ. Файлы QCOW2 могут образовывать цепочки или деревья, в которых только конечные файлы могут служить образами дисков виртуальных систем (это листья дерева, или первые файлы цепочки ссылок, т.е. файлы, на которые нет ссылок). Деревья снимков виртуальных дисков удобно использовать, когда нужно быстро развернуть набор тестовых виртуальных ОС с похожим содержимым, например, в целях функционального тестирования. Внутренние снимки формата QCOW2 менее удобны для

администрирования, и поэтому в данной статье они не рассматриваются.

## Способы создания и хранения снимков виртуальных дисков

Простейший способ формирования инкрементального архива состоит в периодическом создании внешних снимков состояния виртуального диска. Тогда цепочки снимков образуют набор файлов, которые могут быть скопированы в архив (рис.1а). Этот способ имеет несколько очевидных недостатков: (1) избыточный объем данных, поскольку данные об изменениях хранятся в двух экземплярах; (2) использование цепочки снимков замедляет дисковые операции системы виртуализации; (3) большое число QCOW2 файлов в системе виртуализации усложняет администрирование и повышает риск случайной потери данных.



Рис. 1. Процесс архивирования: (а) с двумя параллельными цепочками снимков виртуальных дисков, либо (б) с переносом изменений со времени последнего архивирования; {диск} – текущий снимок, в который выполняется запись данных виртуальной ОС; {0} – начальное состояние дискового устройства. Блоки с двойной рамкой обозначают самодостаточный виртуальный диск, с одинарной – снимок диска, ссылающийся на другой снимок или диск.

Достаточно просто выполнять архивирование состояния диска, создавая только временные снимки в работающей системе виртуализации, но помещая в архив

такую же цепочку инкрементальных снимков (рис.1б). Более подробно такой процесс архивирования изложен ниже (см. пункт "Два способа безостановочного архивирования виртуальной ОС"). Базовый образ виртуального диска не обязан быть в формате QCOW2, это может быть файл в формате RAW, LVM-том или блочное устройство. Таким образом, вне периодов архивирования гостевая система имеет возможность использования наиболее скоростного режима ввода-вывода.

Таким образом, архив состояний виртуального диска состоит из цепочки QCOW2-файлов, где базовый образ – это начальное состояние, самый старый снимок состояния диска. Этот формат архива может быть дополнен двумя простыми идеями:

1. Лучше хранить цепочку файлов с обратным порядком: базовый образ, хранящий последнее состояние диска и историю всех прошлых изменений в виде снимков в формате QCOW2. Такое хранение повышает надежность, так как случайное повреждение или удаление одного инкрементального архива нарушает целостность более старых снимков, а не более новых. Кроме того, повышается скорость доступа к недавно созданным снимкам, а именно они бывают чаще всего нужны.
2. В процессе пополнения цепочки снимков некоторые снимки нужно удалять. Например, если политика архивирования настроена на хранение семи ежедневных и четырех еженедельных архивов, то при создании каждого нового снимка нужно удалять более старый снимок того же типа из цепочки. Удаление снимка не должно затронуть целостность остальных снимков. Эта задача решается с помощью операции 'commit' инструмента qemu-img.

### Обращение порядка цепочки снимков виртуальных дисков

Для обращения порядка цепочки снимков было бы достаточно одной функции "продвижения" базового образа в двунаправленной цепочке файлов (рис. 2). Эту операцию можно реализовать с помощью нескольких операций 'create', 'rebase', и 'commit' инструмента qemu-img (рис. 3):

1. Создается снимок {2}, как ссылка на {диск} образ.

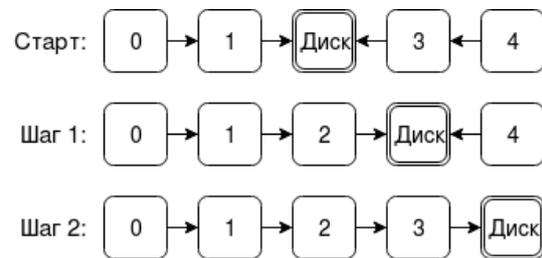


Рис. 2. Операции сдвига базового образа в двунаправленной цепочке снимков. За два шага полный образ диска сдвигается, и становится последним в цепочке архивных копий виртуального диска.

2. Снимок {1} переключается на снимок {2}. Это операция принудительного переключения ('unsafe rebase') утилиты qemu-img, она только переключает ссылку на базовый образ и не проверяет соответствия данных.
3. Снимок {2} переключается на снимок {3}. Это операция 'rebase' утилиты qemu-img. Во время этого шага пустой снимок {2} заполняется старыми версиями данных для тех блоков, которые хранятся как измененные в снимке {3}. Снимок {2} продолжает хранить состояние диска, идентичное {диск}, но теперь как набор изменений содержимого снимка {3}.
4. Выполняется слияние снимка {3} и образа {диск}. Это операция 'commit'. Все изменения диска, записанные в {3} переносятся в образ {диск}, который, фактически, становится снимком {3}.
5. Снимок {4} принудительно переключается на образ {диск}. Снимок {2} принудительно переключается на образ {диск}. Файл снимка {3} удаляется, как ненужный.

Кроме шагов 3 и 4 все операции данного алгоритма элементарны, т.е. выполняются за время  $O(1)$ . Если объем образа {диск} обозначить как  $N$ , а типичный объем инкрементальных изменений диска как  $K$ , то шаги 3 и 4 должны иметь асимптотику  $O(K)$ , поскольку манипуляции выполняются только с теми блоками, которые были изменены. К сожалению, текущая версия утилиты qemu-img выполняет операцию 'rebase' примитивно, путем сравнения всех, даже общих блоков, т.е. без анализа объема реально требуемых действий. Поэтому шаг 3 выполняется за время  $O(N)$ .

Поскольку типичные значения  $K$  меньше  $N$  на 1-3 порядка, этот недостаток утилиты `qemu-img` препятствует эффективной реализации предложенной схемы хранения инкрементальных архивов.

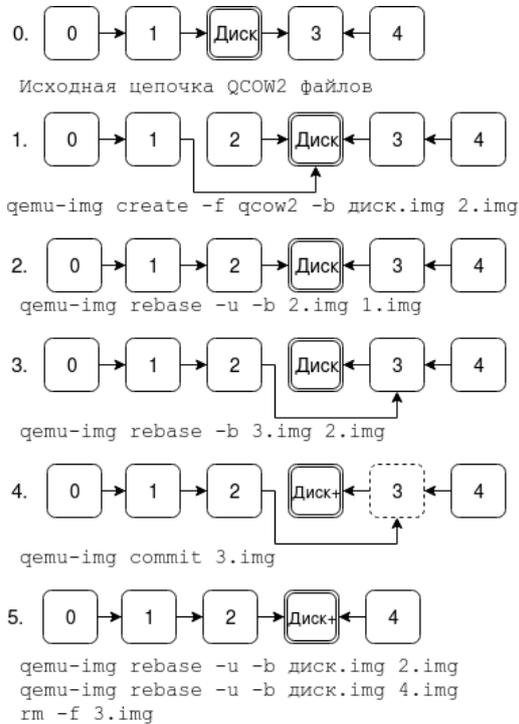


Рис. 3. Последовательность шагов для сдвига базового образа в двунаправленной цепочке снимков.

Рассмотрим подробнее шаг 3. Первоначально снимок {2} не имеет измененных блоков, поскольку он создан как точная копия образа {диск}. В результате операции 'rebase' его содержимое с точки зрения конечного представления данных не должно измениться, но опираться он должен уже на снимок {3}. Поэтому снимок {2} будет содержать ту же самую карту измененных блоков, что и снимок {3}, только не новое, а старое содержимое этих блоков. Тогда снимок {2} будет отменять изменения, представленные снимком {3} по сравнению с образом {диск}. Значит, чтобы выполнить эту операцию, не обязательно прочитывать все блоки образа {диск}, можно пропускать блоки, которых нет в карте снимка {3}. Однако команда `qemu-img rebase` не анализирует карты блоков никаких файлов, кроме самого переключаемого снимка, и это не позволяет ей применить такую очевидную оптимизацию. В общем случае, снимки могут образовывать произвольное дерево, и если операция 'rebase' выполняется для

переноса снимка с одной ветки дерева на другую, то нужно рассматривать только те блоки, которые не являются общими для всех узлов цепочек, соединяющих старое и новое положения снимка в дереве (рис. 4).



Рис. 4. Перенос снимка 1.3.2 => 1.2.1: нет необходимости рассматривать блоки снимка 1, которые не были модифицированы в снимках узлов пути (снимках 1.2 и 1.3).

К сожалению, стиль кода программы `qemu-img` затрудняет выполнение такой обобщенной оптимизации. Но для вышеописанной операции сдвига достаточно проверить, что конечный снимок имеет тот же базовый файл, что и переключаемый снимок, и включить пропуск сравнения общих блоков. Требуемое изменение кода С-программы оказалось небольшим, и модернизированная версия `qemu-img` показала свою эффективность и надежность.

## Удаление снимков

Удаление одного или несколько подряд идущих снимков из цепочки инкрементальных архивов эффективно выполняется одной операцией 'commit' инструмента `qemu-img`.

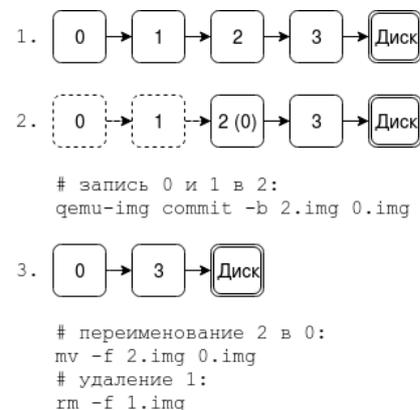


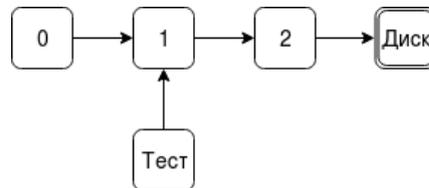
Рис. 5. Операция удаления снимков {1} и {2} из цепочки инкрементальных архивов виртуального диска.

Например, на рис. 5 приведены команды для удаления двух снимков диска. Для удобства можно также восстановить дату и время модификации файла снимка 0 (иначе он будет выглядеть измененным) с помощью команды вида `touch -t <сохраненное значение> 0.img`. С другой стороны, содержимое файла `0.img` изменилось, поэтому устаревшее значение его модификации нарушит работу автоматической синхронизации архивов (например, команды `rsync`), если она не выполняет сверку контрольных сумм, а находит модифицированные файлы по их временной отметке. Чтобы избежать этой проблемы, достаточно увеличить время модификации файла на 1 минуту, тогда файл снимка будет по времени приблизительно соответствовать моменту создания архива, но для системы синхронизации архивов он будет считаться модифицированным и его новое содержимое будет передано получателю. Недостатком внешних снимков является отсутствие встроенной защиты целостности цепочек: модификация, повреждение или удаление файла из цепочки снимков влечет за собой потерю целостности всех снимков, прямо или косвенно опирающихся на этот файл. Попытка использования поврежденных таким образом снимков скорее всего приведет к явной ошибке, но может и не быть вовремя замечена, и тогда данные, извлекаемые из архива, окажутся неверными. Более того, если окажутся повреждены служебные структуры файловой системы, расположенной на виртуальном диске, то потеря данных усугубится поломкой файловой системы и тогда даже частичное восстановление данных окажется непростой задачей.

Файлы цепочки снимков могут быть защищены от случайного удаления или модификации с помощью атрибута `'immutable'` в файловой системе Linux. Кроме того, целостность всей цепочки легко проверять, сохранив самый первый снимок цепочки в виде отдельного файла с полным содержимым, а затем сравнивая эти два виртуальных диска операцией `qemu-img compare`. Либо, чтобы избежать хранения содержимого копии всего диска, хранить только контрольную сумму этих данных. Тогда для проверки целостности нужно будет сначала конвертировать снимок в файл в формат RAW, а затем сверить его содержимое с сохраненной ранее контрольной суммой.

## Использование архивов для восстановления данных

Для доступа к данным архива достаточно создать временный диск, опирающийся на снимок с требуемым состоянием диска (рис. 6).



```
qemu-img create -f qcow2 -b 1.img Тест.имг
```

Рис. 6. Тестовый запуск виртуальной ОС с сохраненным ранее состоянием.

В этом случае тестовая виртуальная ОС запускается с диском {тест}, и все изменения диска останутся в этом снимке, то есть данные цепочки архива не изменятся. Размер временного диска {диск} будет небольшим – для запуска ОС из архивного состояния не требуется много дискового пространства.

## Два способа безостановочного архивирования виртуальной ОС

Самый простой способ использовать снимки дисков гостевой ОС для создания инкрементальных архивов состоит в том, чтобы накапливать изменения содержимого диска с момента предыдущего архивирования, и затем, создав следующий снимок, перенести файл с накопленными изменениями в архив (рис. 7). Особенностью этого метода является то, что в обычном режиме виртуальная ОС работает не с базовым образом, а с файлом QCOW2, опирающимся на базовый виртуальный диск. Это снижает быстродействие дисковых операций по сравнению с работой с "сырыми" устройствами. Преимущество этого подхода в том, что в нем нет долгой операции сравнения содержимого всего диска – копируются только измененные данные. Здесь важно явное предположение о том, что базовый образ в системе виртуализации всегда имеет точно то же содержимое, что и последний снимок архива. Если это условие нарушено, то при очередном сеансе архивирования операция `'unsafe rebase'` приведет к порче данных архива, и все последующие состояния архива будут неверны.

Например, если после шага 4 (рис. 7) у работающей системы был удален и снова создан снимок {диск} (т.е. "слит" с базовым образом диска {3} операцией 'blockcommit', и затем снова создан командой 'virsh snapshot-create-as...'), то дальнейшее архивирование по этому алгоритму будет некорректно.

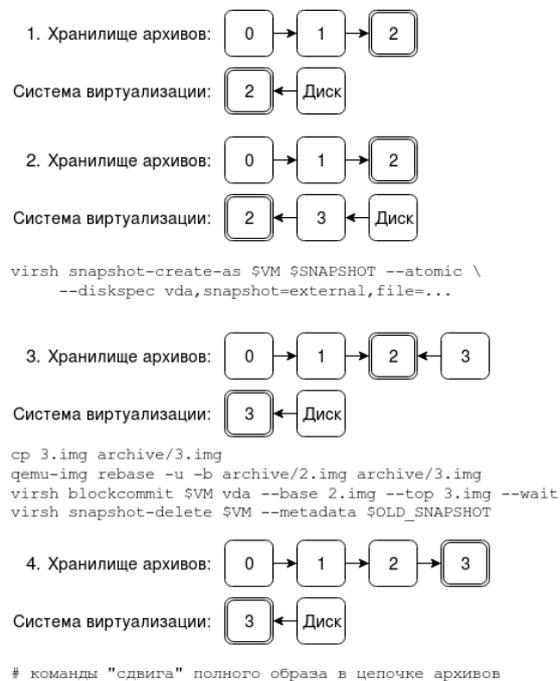


Рис. 7. Процесс архивирования виртуальной ОС (\$VM) с переносом последнего снимка (быстрый, но менее надежный способ, виртуальная ОС все время работает со снимком).

Чтобы избежать такой поломки, достаточно проверять значения временных отметок изменения файлов архива и базового образа диска, и в случае несовпадения выполнить восстановление архивной копии, т.е. сделать полное сравнение базового образа диска с архивом и создать правильный файл изменений.

Другой способ архивирования, лишенный вышеупомянутых недостатков, приведен на рис. 8. В этом методе снимок работающей виртуальной ОС создается только на время архивирования, чтобы зафиксировать состояние диска. Операция 'rebase', прикрепляющая снимок {3} к цепочке архива, требует чтения полного содержимого архивной и рабочей версии диска, и является самой долгой операцией этого метода.

Тем не менее, виртуальная ОС в обоих этих методах недоступна только во

время создания снимка, на шаге 2. В этот момент виртуальная ОС приостановлена для сохранения ее оперативной памяти (если это требуется) и сброса буферов ввода-вывода. В зависимости от параметров виртуальной ОС и метода ее архивирования приостановка занимает от долей секунды до нескольких минут.

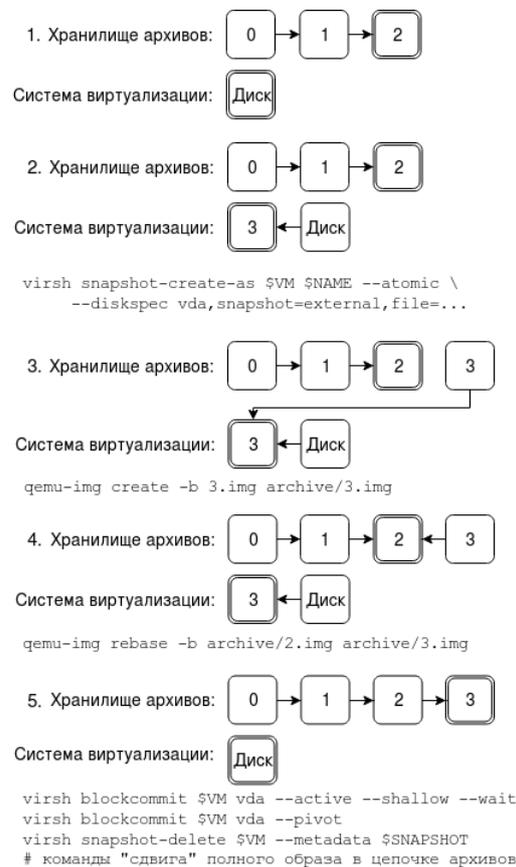


Рис. 8. Процесс архивирования виртуальной ОС (\$VM) с нахождением изменений в процессе архивирования.

В процессе выполнения шага 3 виртуальная ОС уже доступна для пользователей, хотя скорость ее работы несколько снижена, поскольку при этом используется запись во временный снимок диска, а не в базовый образ диска, и поскольку параллельная работа команды 'rebase' дополнительно нагружает подсистему ввода-вывода.

## Обсуждение

Выбор современной универсальной платформы виртуализации обычно сводится либо к коммерческим продуктам компаний VmWare Inc. или Microsoft Inc., либо к набору программ с открытым исходным кодом QEMU/KVM или Xen в среде Linux.

Выбрав коммерческие продукты VmWare vSphere или Microsoft Hyper-V, пользователь сталкивается с тем, что для их эксплуатации (в частности, для организации архивирования виртуальных ОС) необходимы дополнительные программы сторонних компаний. Эти программы либо дорогостоящие, либо набор их функций сильно ограничен.

Компания RedHat, конкурирующая в области виртуализации с компаниями VmWare Inc. и Microsoft Inc., предлагает продукт RedHat Enterprise Virtualization (RHEV) и услугу поддержки его пользователей, но в то же время по лицензии GPL распространяет идентичный продукт oVirt, который поддерживается по модели открытого сообщества разработчиков. Продукты RHEV и oVirt основаны на технологиях QEMU/KVM и позволяют использовать кластер вычислительных машин для эксплуатации набора виртуальных ОС. К сожалению, полноценной функции архивирования виртуальных ОС в этом продукте пока нет, она находится в стадии разработки и запланирована в будущих релизах RHEV/oVirt [4].

В данной работе приведена стратегия и методы для инкрементального архивирования виртуальных ОС в среде QEMU/KVM/Libvirt, решающие три задачи: архивирование без остановки виртуальных ОС, экономность хранения архивов и контроль сохранности архивов. Для решения этих задач были использованы следующие средства: атомарные безостановочные снимки (функция QEMU/libvirt) и инкрементальные архивы виртуальных дисков, образующие цепочки внешних снимков (QCOW2-формат QEMU, и утилита qemu-img). В процессе тестирования этих решений и эксплуатации виртуальных ОС были замечены некоторые недостатки и особенности использованных средств:

- нестабильность работы службы агента виртуализации QEMU для Windows. Эта служба аварийно останавливалась после первого архивирования диска методом "freeze" (данная проблема описана в базе данных ошибок RedHat под номером 1387125, где описан метод ее решения);
- некоторые прикладные системы в ОС Windows не могут быть настроены для завершения транзакций и сброса буферов на диск, необходимого для архивирования

методом "freeze". Поэтому для виртуальных систем Windows пришлось использовать метод архивирования "memory snapshot";

- функция 'rebase' утилиты qemu-img не пропускает сравнение блоков данных, когда заранее известно, что данные этих блоков не могут различаться. Предложенное исправление кода утилиты qemu-img устранило этот недостаток для тех случаев, когда легко определить общий базовый образ.

Данные проблемы были частично решены, хотя они не являлись препятствием для регулярного архивирования виртуальных ОС в период неактивности прикладных систем (например, вне рабочего времени). Если требуется частое архивирование нагруженных систем, например, каждый час, то без заметного перерыва в доступности этого пока можно достичь в среде QEMU/KVM только при архивировании методом "freeze", то есть без снимка оперативной памяти. При этом необходимо будет также ограничить нагрузку на подсистему ввода-вывода процесса архивирования, работающего в фоне (в системе Linux лимит на скорость ввода-вывода устанавливается с помощью инструмента cgroups).

Создание снимка с полным сохранением памяти и состояния устройств пока требует относительно длительной приостановки виртуальной ОС. Тем не менее разработчики системы QEMU обсуждают использование новой функции ядра userfaultfd для ускорения миграции виртуальных ОС с одной хост-системы на другую [5,6] и для ускоренного создания снимка оперативной памяти [7]. Эта функция позволяет сделать мгновенный снимок оперативной памяти и далее осуществлять его фоновое сохранение, в то время, как виртуальная ОС продолжит свою работу, изменяя страницы памяти в режиме "copy-on-write".

Так можно существенно ускорить сохранение полного состояния виртуальной ОС. В сентябре 2018 года компания Huawei Technologies Inc. получила патент [8] на такой алгоритм создания снимка виртуальной ОС, и теперь этот патент является юридическим препятствием для использования userfaultfd в проекте QEMU/KVM. Хотя разработчик QEMU/KVM, который предлагал это решение, Zhang Hailiang, на момент обсуждения являлся сотрудником Huawei Technologies Inc. (судя по его e-mail адресу) и, возможно, политика этой компании не будет препятствовать

использованию данного патента в некоммерческих программах.

## Заключение

Среда виртуализации QEMU/KVM/Libvirt в ОС Linux, построенная по принципам открытой архитектуры, имеет достаточный набор функций, чтобы эффективно решить задачу построения инкрементального архива виртуальных операционных систем. Открытость кода всех инструментов этой среды дает пользователям богатые возможности по ее настройке, а также позволяет при необходимости исправлять выявленные недостатки.

Предложенные в данной статье решения были применены для архивирования различных виртуальных операционных систем с архитектурами Linux и Windows. Архивирование выполнялось методами "memory snapshot" и "freeze" с хранением цепочки инкрементальных архивов и на протяжении более чем двух лет непрерывной эксплуатации показало свою эффективность и надежность. Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (выполнение фундаментальных научных исследований ГП 14) по теме № 0065-2018-0004 "Развитие методов математического моделирования распределенных систем и соответствующих методов вычисления." (№ АААА-А18-118041290146-7).

# Optimal backup strategies in GNU Linux/QEMU/KVM virtualization environment

A.B. Betelin, A.A. Prilipko, G.A. Prilipko, S.G. Romanyuk, D.V. Samborskiy

**Abstract:** QEMU/KVM/Libvirt virtualization environment allows creation of virtual machine (VM) live snapshots. This article introduces optimal methods for generation and storage of incremental backups for QEMU/KVM virtual machines. The suggested techniques optimize speed, storage space, and archive reliability. Two-year experience of maintaining VM incremental backups showed that when used correctly, QEMU/KVM/Libvirt tools correspond by the basic criteria to the main commercial products in this area.

**Keywords:** Virtualization, Linux, KVM, QEMU, Libvirt, copy-on-write, incremental backup, userfaultfd.

## Литература

1. Сайт проекта KVM. <https://www.linux-kvm.org/page>. Дата обращения 18.10.2018.
2. Сайт "QEMU, the FAST! processor emulator". <https://www.qemu.org>. Дата обращения 18.10.2018.
3. Сайт "Libvirt: The virtualization API". <https://libvirt.org>. Дата обращения 18.10.2018.
4. Сайт "oVirt Feature List: Incremental Backup". <https://ovirt.org/development/releasemanagement/features/storage/incremental-backup>. Дата обращения 18.10.2018.
5. Mike Rapoport. "Userfaultfd: Post-copy VM migration and beyond", [https://blog.linuxplumbersconf.org/2017/ocw/system/presentations/4699/original/userfaultfd\\_post-copy\\_VM\\_migration\\_and\\_beyond.pdf](https://blog.linuxplumbersconf.org/2017/ocw/system/presentations/4699/original/userfaultfd_post-copy_VM_migration_and_beyond.pdf). Дата обращения 18.10.2018.
6. Yoshihisa Abe, Roxana Geambasu, Kaustubh Joshi, Mahadev Satyanarayanan. "Urgent Virtual Machine Eviction with Enlightened Post-Copy". VEE 2016, Proceedings of the 12th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, pages 51-64.
7. Andrea Arcangeli, Zhang Hailiang. Обсуждения "[Qemu-devel] [RFC 00/13] Live memory snapshot based on userfaultfd", <https://lists.gnu.org/archive/html/qemu-devel/2016-01/msg00664.html> и "[Qemu-devel] [PATCH 00/17] RFC: userfault v2" <https://lists.nongnu.org/archive/html/qemu-devel/2014-11/msg00020.html>. Дата обращения 18.10.2018.
8. Huawei Technologies. Патент US 20180276081A. "Method and apparatus for generating virtual machine snapshot".
9. Сайт проекта Xen. <https://xenproject.org>. Дата обращения 18.10.2018.
10. А.О. Кудрявцев, В.К. Кошелев, А.О. Избышев, И.А. Дудина, Ш.Ф. Курмангалеев, А.И. Аветисян, В.П. Иванников, В.Е. Велихов, Е.А. Рябинкин. "Разработка и реализация облачной системы для решения высокопроизводительных задач". Труды Института системного программирования РАН, выпуск 24 (2013).

# Методы и средства оценки объема памяти, требуемой для выполнения образа ОСРВ Багет

А.Н. Годунов<sup>1</sup>, В.А. Солдатов<sup>2</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия, E-mail's: <sup>1</sup>nkag@niisi.ras.ru, <sup>2</sup>nkvalera@niisi.ras.ru

**Аннотация:** Существует класс задач, работающих в жестком реальном времени, для которых оперативная память является лимитированным ресурсом. Разработка методов сокращения объема памяти, используемой ОСРВ, предполагает наличие отладочного стенда и инструментальных средств количественной оценки объема памяти, требуемой для выполнения полученного образа. Всю память, требуемую выполняемому образу, можно условно разбить на статическую и динамическую. Под статической памятью понимается память, требуемая программному коду и сегментам данных образа. Для оценки объема требуемой статической памяти предлагается использовать специально разработанную утилиту, которая выполняется на инструментальной ЭВМ, и позволяет оценить сколько статической памяти требуется всего, и сколько памяти требуется отдельным частям образа. К динамической памяти относится память, которая выделяется программе уже при ее выполнении. Для оценки требуемого объема динамической памяти разработана соответствующая функция и указаны точки ее применения. Описан отладочный стенд, созданный специально для разработки компактной операционной системы.

**Ключевые слова:** ОСРВ, масштабирование, конфигурирование, статическая память, динамическая память, формат ELF, ОСРВ Багет.

## Введение

Для некоторого класса задач, которые выполняются в жестком реальном времени, оперативная память является лимитированным ресурсом. В первую очередь это относится к системам, работающим в условиях радиационного облучения, т.к. радиационно-стойкая оперативная память является достаточно дорогой. В данной работе считается, что максимальный объем оперативной памяти ограничивается 4 Мбайтами. Кроме оперативной памяти имеется еще и постоянное запоминающее устройство (флэш-память), которое может использоваться для хранения неизменяемых данных.

В [1] были описаны основные методы, которые применяются при создании компактной ОСРВ на основе уже существующей ОС. В качестве базовой ОС была выбрана ОСРВ из семейства Багет 2.x [2,3], которую в дальнейшем будем называть ОСРВ Багет. Этот выбор был обусловлен следующими основными причинами:

- достаточно длительной историей существования ОС (первый выпуск состоялся в 2001 году);
- непрерывным развитием (всего выпущено 6 изданий);
- широким распространением (используется более чем в 100 организациях);

- удовлетворение требованиям информационной безопасности и высокой надежности (тестирование независимой системой тестов для операционных систем ИСП РАН [4]).

Там же ([1]) были сформулированы основные методы для сокращения объема используемой ОСРВ Багет оперативной памяти:

- опция оптимизации компилятора –Os, которая позволяет сократить объем объектных файлов;
- размещение программных сегментов целевого образа во флэш-памяти;
- масштабирование (scaling).

Для оценки результатов, полученных с помощью разработанных методов, предлагается использовать средства, описываемые в данной работе.

Эти средства позволяют оценивать потребности в памяти всего образа или его отдельных частей.

Практическая проверка результатов по сокращению требований к используемому объему памяти производилась на отладочном стенде, состоящим из:

- инструментальной ЭВМ с операционной системой LINUX;
- целевой ЭВМ с радиационно-стойким микропроцессором.

## Формат объектного файла OCPB

Образ OCPB Багет совместно с ППМ и прикладными программами представляет собой объектный исполняемый файл в формате ELF. В этом файле хранится программа, которая будет запущена на целевой ЭВМ; все адреса этой программы уже настроены на соответствующее место памяти. Рассмотрим кратко формат объектного файла, останавливаясь в основном на понятиях, используемых OCPB Багет.

Файл состоит из заголовков и разделов (сегментов).

В самом начале файла расположен заголовок ELF, который описывает структуру файла и содержит следующие данные:

- описатель файла;
- тип объектного файла;
- архитектура аппаратной платформы (Intel, MIPS и т.п.);
- виртуальный адрес точки входа в программу;
- месторасположение таблицы заголовков программы;
- месторасположение таблицы заголовков разделов (сегментов);
- месторасположение раздела, с таблицей строк, которая содержит в себе имена разделов и символов объектных файлов.

В идентификаторе файла содержатся: сигнатура, определяющая файл типа ELF; номер версии формата ELF; архитектура системы (32-х или 64-х разрядная, порядок следования байт и т.п.).

Тип объектного файла может принимать следующие значения: перемещаемый, разделяемый или исполняемый. Нам в первую очередь интересуют исполняемые объектные файлы, которые используются загрузчиком. Перемещаемые объектные файлы в OCPB Багет используются динамическим загрузчиком. Разделяемые объектные файлы в OCPB Багет не используются.

Таблица заголовков программы содержит информацию, необходимую для загрузки кода образа в память.

Каждый элемент таблицы заголовков разделов описывает один раздел и содержит следующую информацию:

- тип сегмента и действия операционной системы с данным сегментом;
- расположение сегмента в файле;
- стартовый адрес сегмента в виртуальной памяти образа;
- размер сегмента в файле;
- размер сегмента в памяти ЭВМ;
- флаги режима доступа к сегменту (запись, чтение, выполнение).

Некоторые специальные разделы могут содержать управляющую и программную информацию. Рассмотрим основные из них.

«bss» и «sbss» – в разделах с этими именами хранятся неинициализированные данные, относящиеся к образу памяти в программе. Эти разделы не занимают места в объектном файле. В разделе с именем «sbss» хранятся короткие данные.

«data» и «sdata» – в разделах с этими именами хранятся инициализированные данные, относящиеся к образу памяти в программе. В разделе с именем «sdata» хранятся короткие данные.

«.rodata» – в разделе с этим именем хранятся данные, предназначенные только для чтения.

«.text» – в разделе с этим именем хранится текст (исполняемые инструкции) программы. При загрузке этот сегмент может быть размещен не только в оперативной, но и во флэш-памяти.

«.symtab» – в разделе с этим именем хранится таблица имен.

«.comment» – в разделе с этим именем хранится дополнительная информация.

В разделах, имена которых начинаются с префикса «.debug», хранится информация для отладки.

Объем памяти, который требуется для выполнения любой программы, можно условно разделить на следующие три части:

- память, в которой размещается программный код и неизменяемые константы программы;
- память, в которой размещаются глобальные и статические данные, используемые программой;
- память, которая динамически выделяется программе при ее выполнении.

Первый вид памяти из этого списка используется только для чтения, и тем самым сегменты образа OCPB Багет, которые не имеют флага, разрешающего запись, могут быть размещены как в оперативной памяти, так и во флэш-памяти.

Память первых двух видов из списка будем называть статической памятью, а третий вид памяти из этого же списка – динамической памятью. Сколько статической памяти требуется для выполнения программы можно оценить до ее выполнения путем анализа объектного кода программы.

Для оценки размера динамической памяти, используемой программой, можно использовать различные методы, но все они могут не дать точного результата. Метод, предлагаемый в работе, будет рассмотрен позже.

## Оценка размера статической памяти образа ОСРВ

Для оценки размера оперативной памяти, которая потребуется для загрузки образа ОСРВ на целевую ЭВМ без учета памяти, которая будет запрошена динамически после начала выполнения образа, можно воспользоваться утилитой «gsc objdump». Пример такой выдачи представлен в таблице 1.

В начале выводится заголовок таблицы, в котором содержится имя файла (ос2000) и формат файла (elf32-big). Далее в виде таблицы выводится информация о сегментах файла. Колонки в этой таблице имеют следующий смысл:

Idx – порядковый номер сегмента;

Name – имя сегмента;

Size – размер сегмента в байтах;

VMA (virtual memory address) – виртуальный адрес сегмента в памяти;

LMA (load memory address) – загрузочный адрес сегмента в памяти;

File off – адрес начала сегмента в файле;

Algn – величина выравнивания адреса начала сегмента в памяти;

Flags – флаги, определяющие свойства сегмента.

Флаги, представленные в последней колонке, имеют следующий смысл:

CONTENTS – сегмент содержит данные;

ALLOC – сегмент занимает память при выполнении на целевой ЭВМ; если у сегмента нет этого флага, то он не хранится в памяти целевой ЭВМ;

```
ос2000: file format elf32-big
```

```
Sections:
```

Idx Name	Size	VMA	LMA	File off	Algn	Flags
0 .text	001473b8	80200000	80200000	00001000	2**12	CONTENTS, ALLOC, LOAD, CODE
1 .data	0001693c	803473b8	803473b8	001483b8	2**3	CONTENTS, ALLOC, LOAD, DATA
2 .bss	0000c638	8035dcf8	8035dcf8	0015ecf4	2**3	ALLOC
3 .reginfo	00000018	00000000	00000000	0015ecf4	2**2	CONTENTS, READONLY
4 .pdr	000166c0	00000000	00000000	0015ed0c	2**2	CONTENTS, READONLY
5 .debug_abbrev	0003327d	00000000	00000000	001753cc	2**0	CONTENTS, READONLY, DEBUGGING
6 .debug_info	002878f7	00000000	00000000	001a8649	2**0	CONTENTS, READONLY, DEBUGGING
7 .debug_line	00059a22	00000000	00000000	0042ff40	2**0	CONTENTS, READONLY, DEBUGGING
8 .comment	00004448	00000000	00000000	00489962	2**0	CONTENTS, READONLY
9 .debug_aranges	00002f20	00000000	00000000	0048ddb0	2**3	CONTENTS, READONLY, DEBUGGING
10 .debug_pub	0000b4c6	00000000	00000000	00490cd0	2**0	CONTENTS, READONLY, DEBUGGING
11 .debug_frame	000124b4	00000000	00000000	0049c198	2**2	CONTENTS, READONLY, DEBUGGING
12 .debug_str	00005440	00000000	00000000	004ae64c	2**0	CONTENTS, READONLY, DEBUGGING
13 .debug_loc	000b8867	00000000	00000000	004b3a8c	2**0	CONTENTS, READONLY, DEBUGGING
14 .debug_ranges	0001df88	00000000	00000000	0056c2f3	2**0	CONTENTS, READONLY, DEBUGGING

Таблица 1. Информации о размерах статической памяти (objdump)

LOAD – данным сегмента, переносимым в память целевой ЭВМ, выделяется непрерывный участок виртуальной памяти;

CODE – данные сегмента содержат исполняемый код;

READONLY – данные сегмента предназначены только для чтения;

DEBUGGING – данные сегмента содержат информацию для отладки.

В ОСРВ Багет на целевую ЭВМ загружаются только сегменты, имеющие флаг ALLOC.

Сегмент “.text” из таблицы 1 может быть размещен как в оперативной памяти (RAM), так и во флэш-памяти (ROM), сегменты “.data” и “.bss” могут быть размещены только в оперативной памяти, остальные сегменты на целевую ЭВМ не загружаются. Таким

образом, в этой таблице содержится вся информация, требуемая для оценки объема памяти, необходимой для размещения сегментов программного кода и данных на целевой ЭВМ. Однако формат представленной информации не содержит общего объема памяти, и все числа представлены только в шестнадцатеричном формате. Поэтому был разработан сценарий (\_ос2000\_seg.sh), который преобразует выдачу утилиты objdump к виду, представленному в таблице 2.

Таблица состоит из двух частей. В первой части таблицы выводится заголовок “ELF sections:”, далее построчно выводится информация о сегментах, которые загружаются на целевую ЭВМ. Для каждого сегмента указываются:

– порядковый номер;

- имя сегмента (seg=<имя>);
- размер сегмента в шестнадцатеричном и десятичном форматах (size=);
- тип памяти, пригодный для размещения сегмента.

```

ELF section:
0. seg=.text: size=001473b8 (1340344):
                                mem=ROM
1. seg=.data: size=0001693c ( 92476):
                                mem=RAM
2. seg=.bss: size=0000c638 ( 50744):
                                mem=RAM

Total Memory Usage:
MEM size = 2048 Kb (100.0%)
RAM seg  = 139 Kb ( 6.8%)
ROM seg  = 1308 Kb (63.9%)
MEM usage = 1448 Kb (70.7%)
MEM free  = 599 Kb (29.3%)

```

Таблица 2. Информация о размерах статической памяти образа (размещение в RAM)

Вторая часть таблицы также начинается с заголовка “Total Memory Usage:” (Использование памяти), за которым последовательно выводятся:

- размер всей оперативной памяти в Кб, который принимается за 100%;
- суммарный размер всех сегментов, которые могут быть размещены только в оперативной памяти и процент по отношению к размеру всей оперативной памяти;
- общий размер сегментов, которые могут быть размещены как в оперативной памяти, так и во флэш-памяти, и процент по отношению к размеру всей оперативной памяти;
- размер оперативной памяти, требуемой для размещения загружаемых сегментов целевого образа, и процент по отношению к размеру всей оперативной памяти;
- размер оперативной памяти, оставшейся свободной после загрузки целевого образа, и процент по отношению к размеру всей оперативной памяти.

У рассматриваемого сценария имеются два позиционных параметра.

Первый параметр определяет, куда предполагается помещать сегменты, которые могут быть размещены не только в оперативной памяти (1 – в оперативную память, 2 – во флэш-память).

Второй параметр задает общий объем оперативной памяти в мегабайтах. В данном примере первый параметр равен 1, а второй 2, что соответствует значениям по умолчанию сценария.

```

ELF section:
0. seg=.text: size=001473b8 (1340344):
                                mem=ROM
1. seg=.data: size=0001693c ( 92476):
                                mem=RAM
2. seg=.bss: size=0000c638 ( 50744):
                                mem=RAM

Total Memory Usage:
MEM size = 2048 Kb (100.0%)
RAM seg  = 139 Kb ( 6.8%)
ROM seg  = 1308 Kb (63.9%)
MEM usage = 139 Kb ( 6.8%)
MEM free  = 1908 Kb (93.2%)

```

Таблица 3. Информация о размерах статической памяти образа (размещение в RAM и ROM)

Если первый параметр задать равным 2, тогда мы получим вид, представленный в таблице 3.

Таким образом с помощью сценария `_oc2000_seg.sh` можно оценивать общий объем статической памяти, используемый конкретным образом системы.

Для оценки размера оперативной памяти, которая требуется для каждого объектного модуля, можно использовать опцию редактора связей `gcc -print-map`, которая предписывает выводить подробную информацию о каждом объектном модуле, входящем в образ системы. Другими словами, создавать карту образа (`map`). Информация, выдаваемая редактором связей, состоит из нескольких различных частей, следующих друг за другом. В начале каждой части выводится ее заголовок.

Первая часть выдачи содержит список модулей, включенных в собираемый образ для разрешения внешних ссылок. Каждый элемент списка состоит из двух строк. В первой строке содержится имя модуля, который включается в образ, а во второй – какая именно внешняя ссылка и какого модуля требует включить его в собираемый образ. Фрагмент выдачи представлен в таблице 4.

В случае, когда в образе оказался модуль, который не должен там содержаться, из соответствующей строки списка легко увидеть, какой модуль и какая внешняя ссылка вызвали эту ситуацию.

В последней части выводится скрипт, который управлял сборкой образа и карта памяти, состоящая из списка включенных сегментов. Для каждого сегмента указывается его смещение в целевом образе, размер и имя модуля, из которого взят сегмент. Если сегмент модуля содержит внешние имена,

тогда для каждого из них указывается смещение в целевом образе (таблица 5).

Карта памяти содержит всю информацию, необходимую для анализа использования памяти каждым модулем, входящим в целевой образ. Однако формат, в котором представлена эта информация не удобен для зрительного восприятия. Для анализа потребления памяти удобно образ ОС разбить на отдельные части. Наименование части и список модулей, входящих в эту часть, готовится заранее. Например, вводится понятие части с наименованием “shell”, к которой отно-

сятся модули: `syntbl.c`, `shell.c`, `s_y.c`, `s_lex.c`, `alias_func.c`, `help.c`, `sys_shell.c` и `sym_tbl.S`.

По аналогии все модули, входящие в состав образа, должны быть отнесены к какой-либо части.

Если модуль не содержится ни в какой части, тогда он будет отнесен к части PPM.

На основании сказанного была разработана утилита `mini` на языке `java`, которая преобразует карту памяти образа из таблицы 5 в вид удобный для анализа использования статической памяти.

Пример выдачи представлен в таблице 6.

```
Archive member included because of file (symbol)
/home/nkvalera/oc277-005h/lib/bspbt205.a(cpuLib.o)
                                                    oc2000.o (cp0PrIDGet)
/home/nkvalera/oc277-005h/lib/bspbt205.a(cacheR3kLib.o)
                                                    oc2000.o (cacheInit)
.
.
/home/nkvalera/oc277-005h/lib/oc2000mips.a(queue.o)
/home/nkvalera/oc277-005h/lib/oc2000mips.a(q_ddl.o) (queInitDef)
```

Таблица 4. Фрагмент списка модулей, включенных в образ для разрешения внешних ссылок

```
Linker script and memory map
LOAD oc2000.o
.
.
END GROUP
.
.text      0x0000000000000000  0x18698
*(.text .stub)
.text      0x0000000000000000  0x3208 oc2000.o
           0x00000000000000ea4  kvprintf
.
           0x00000000000000cfc  boardWDRReset
.text      0x00000000000003208  0x210 mem_show.o
.
           0x000000000000032b8  start
.text      0x00000000000003418  0x33c cpuLib.o
.
           0x0000000000000182cc  queInitDef
.rodata    0x0000000000000000  0x1c54
*(.rodata)
.rodata    0x0000000000000d40  0x5c /home/nkvalera/oc277-005h/lib/oc2000mips.a(mempool.o)
.
.rodata    0x00000000000001a54  0x200 /home/nkvalera/oc277-005h/lib/oc2000mips.a(fsb.o)
           0x00000000000001a54  fmsbTbl
.
.data      0x0000000000000000  0x588
*(.data)
.data      0x0000000000000000  0x12c oc2000.o
           0x00000000000000014  boardCpuClockRate
```

Таблица 5. Скрипт редактора связей и карта памяти целевого образа

Type	.text	.rodata	.data	.bss	.sdata	.sbss	TOTAL
shell	53080	51657	79252	188	64	52	184293
net	133472	10764	1040	2228	88	285	147877
nfs	134316	2352	600	7392	40	256	144956
os	85660	1508	1640	15368	120	220	104516
mips	27796	76528	0	56	44	36	104460
PPM	85604	12244	104	2004	0	4	99960
netinet	82956	4280	860	3004	148	204	91452
unix	72220	5900	2580	9476	392	328	90896
validator	51456	22116	0	0	0	8	73580
debug	44840	14956	832	392	36	172	61228
msdosfs	54164	2044	464	80	16	68	56836
rpc	38632	2804	296	76	52	26	41886
init_os	26308	5896	872	4928	0	0	38004
stdio	28016	7136	0	0	56	12	35220
fsck	31528	2532	0	0	8	4	34072
cd9660	23348	864	648	0	8	28	24896
term	20016	96	0	0	0	0	20112
remote-debug	16652	2876	16	80	20	68	19712
show	12768	3444	0	0	4	8	16224
stdlib	15060	724	152	128	36	24	16124
mdload	13064	2284	136	160	0	56	15700
utils	12200	2484	0	0	0	0	14684
mtx	10020	148	0	388	4	4	10564
time	9108	432	324	252	12	4	10132
sig	8884	76	396	332	4	4	9696
mq	8852	32	16	340	16	16	9272
prsmem	7276	952	0	0	0	4	8232
gcclib	7812	296	0	0	0	0	8108
libm	5256	128	2064	0	624	0	8072
syslog	7012	192	0	116	16	8	7344
tar	5688	628	224	0	8	4	6552
mem	5128	92	0	984	16	0	6220
sem	5872	28	0	196	4	4	6104
dir	5248	28	0	0	0	0	5276
locale	3408	1612	0	0	0	8	5028
kprint	4528	452	0	0	0	0	4980
cond	3852	132	0	368	0	4	4356
tmr	4132	28	0	160	0	4	4324
fileio	4236	4	0	0	0	0	4240
string	3244	512	0	0	0	4	3760
ieee	3180	344	0	56	0	0	3580
encoding	1216	4	0	0	0	16	1236
pipe	1176	36	0	0	0	0	1212
posix	760	0	0	0	0	0	760
setjmp	740	0	0	0	0	0	740
user	508	0	0	0	0	0	508
wchar	124	0	0	0	0	0	124
<b>TOTAL</b>	<b>1180416</b>	<b>241645</b>	<b>92516</b>	<b>48752</b>	<b>1836</b>	<b>1943</b>	<b>1567108</b>

Таблица 6. Объем статической памяти, требуемой отдельным частям образа

Type	.text	.rodata	.data	.bss	.sdata	.sbss	TOTAL
<b>====nfs:</b>							
nfs_nqlease.o	5256	40	104	0		8	5408
nfs_bio.o	12092	208	0	0			12300
nfs_mount.o	992		0	0			992
nfs_socket.o	13256	852	0	5700	16	20	19844
nfs_node.o	1860	120	0	0		20	2000
nfs_vnops.o	77492	628	272	160	12	20	78584
nfs_subs.o	11444	100	136	0	4	140	11824
krpc_subr.o	2128	120	0	0	4		2252
nfs_syscalls.o	2536	96	0	0		4	2636
nfs_stub.o	32		0	0	4	4	40
nfs_vfsops.o	7228	188	88	1532		40	9076
<b>total nfs</b>	<b>134316</b>	<b>2352</b>	<b>600</b>	<b>7392</b>	<b>40</b>	<b>256</b>	<b>144956</b>
<b>====remote-debug:</b>							
xdr_ld.o	304		0	0			304
rdebug.o	13748	2688	0	80	20	60	16596
xdr_rdb.o	1064	24	0	0			1088
xdr_ptrace.o	436		16	0			452
rdebuga.o	1100	164	0	0		8	1272
<b>total remote-debug</b>	<b>16652</b>	<b>2876</b>	<b>16</b>	<b>80</b>	<b>20</b>	<b>68</b>	<b>19712</b>
<b>====mdload:</b>							
elfload.o	8716	1544	0	0		8	10268
module.o	4348	740	136	160		48	5432
<b>total mdload</b>	<b>13064</b>	<b>2284</b>	<b>136</b>	<b>160</b>		<b>56</b>	<b>15700</b>

Таблица 7. Пример детализации по отдельным частям образа

Колонки «Type» и «Total» в таблице 6 имеют следующий смысл:

Type – наименование части образа;

Total – суммарный размер всех сегментов.

Колонки с наименованием “.text”, “.rodata”, “.data”, “.bss”, “.sdata”, “.sbss” – содержат размер одноименных сегментов в десятичном формате.

Каждая строка таблицы 6 содержит информацию об одной части образа ОСРВ Багет. В конце таблицы выводится итоговая строка.

Утилита `mini` может также выдавать детализацию по одной части образа или по всем частям образа.

В этом случае в колонке Type выводятся наименования модулей, относящихся к части. Например, детализация по частям образа, которые содержат модули nfs клиента, удаленного отладчика (remote debugger) и динамического загрузчика (mdload) представлена в таблице 7.

В этой таблице перед списком модулей, относящихся к части образа, выводится промежуточный заголовок с наименованием части, а в конце списка выводится строка, содержащая объем требуемой памяти всей части образа.

## Оценка размера динамической памяти

Для оценки размера оперативной памяти, которая выделяется динамически, рассмотрим общие принципы распределения памяти. В ОСРВ Багет реализованы функции (динамического) распределения памяти в соответствии со стандартом C, а также реализован механизм распределения памяти на основе так называемых пулов памяти.

К стандартным средствам относятся функции:

- `calloc()` - выделение и очистка памяти;
- `free()` - освобождение памяти;
- `malloc()` - выделение памяти;
- `realloc()` - перераспределение памяти.

Использования перечисленных функций не требует знакомства с пулами памяти (пулы памяти используются неявно). Однако, при порождении потока управления можно указать какой пул памяти будут использовать эти функции в контексте порождаемого потока (см. ниже).

Пул памяти может быть создан функцией `memPoolCreate()` на любом свободном (непрерывном) участке памяти. Непосредственно после создания пула вся его память (за

исключением небольшого участка занятого управляющей информацией) объявляется свободной и может быть выделена прикладной программе или ОС. В дальнейшем к пулу памяти могут быть добавлены другие, вообще говоря, несмежные участки памяти функцией `memPoolAdd()`. Системный пул памяти создается при инициализации системы.

Память, находящаяся в пуле, используется следующим образом. В начале нужно получить участок памяти из пула с помощью функции `memAlloc()`. После получения участка памяти он может использоваться прикладной программой. Когда необходимость в данном участке памяти отпадет, его следует освободить (вернуть в пул памяти) функцией `memFree()`. После этого операционная система рассматривает этот участок памяти как свободный (и может выделить его в результате последующего обращения к функции `memAlloc()`).

При инициализации системы (в функции `osBoardInit()`) определяется начало свободной памяти `osFreeRamAdr`, которая начинается сразу за памятью, занятой программным кодом и сегментами с данными. Далее в функции `kernInit (char *pMemBeg, char *pMemEnd, u_int intStackSize)`, которой в качестве параметров передаются адреса начала и окончания свободной памяти, а также размер стека для обработчика прерываний, производится сдвиг указателя начала памяти (`*pMemBeg`) на размер стека для обработчика прерываний (`intStackSize`) и инициализация системного пула памяти (от `*pMemBeg` до `*pMemEnd`), т.е. вся свободная память отводится под системный пул. Следует также учесть, что в самом начале оперативной памяти в процессорах архитектуры MIPS располагаются векторы обработки исключительных ситуаций. В ОСРВ Багет часть памяти, расположенная после векторов прерываний, отводится под «сохраняемую» память, в которую при выполнении системы на целевой ЭВМ можно записывать различную информацию, и она будет там сохранена при перезагрузке без выключения питания. Распределение памяти представлено на рисунке 1.

В процессе работы можно создавать новые пулы памяти из памяти системного пула или из памяти других ранее созданных пулов:

- с помощью функции `memAlloc()` берется участок памяти требуемого размера из соответствующего пула;
- на полученной памяти с помощью функции `memPoolCreate()` создается новый пул памяти, или с помощью функции

`memPoolAdd()` эта память добавляется к уже существующему пулу.

Векторы исключительных ситуаций
Сохраняемая память
Программный сегмент и сегменты данных исполняемого образа
Стек обработчика прерываний
Системный пул памяти

Рисунок 1. Распределение оперативной памяти

У каждого потока имеется пул памяти, который используется по умолчанию. Этот пул памяти используется функциями стандарта C `calloc()`, `free()`, `malloc()`, `realloc()`, а также функциями `memAlloc()`, `memFree()`, если пул памяти не указан (указано `NULL`). Из этого же пула памяти берется память под стек потока управления (если не указан атрибут `stackaddr`). Используемый по умолчанию пул памяти указывается при порождении потока управления (атрибут `mempool`). Если при порождении потока не указать пул памяти, то будет использоваться системный пул. Использование пулов памяти позволяет избежать фрагментации памяти, а также позволяет уменьшить нежелательное влияние одних потоков на переменные других потоков, возникающее при наличии ошибок в программном коде.

Для оценки памяти, используемой в системе динамически, достаточно учитывать память системного пула. Если в исполняемом образе имеется консоль и включены информационные команды, тогда можно пользоваться командой `mem`, которая предназначена для вывода сведений о распределении памяти. Если выполнить эту команду, то выводится информация о состоянии памяти в настоящий момент, которая содержит: адрес описателя пула памяти, объем распределенной (используемой в настоящий момент) памяти в байтах, количество выделенных блоков, объем свободной памяти в байтах, количество свободных блоков, размер максимального свободного блока.

Для получения информации об использовании памяти можно также обратиться к функции `mem_showS(int *arg)`, разработанной специально для выполняемой работы. В качестве аргумента этой функции передается указатель на массив, состоящий из трех це-

лых чисел. В результате выполнения функции массив, адрес которого был передан, будет содержать: общий размер системного пула, сколько памяти уже выделено, сколько свободной памяти осталось.

Выполняя эту функцию перед началом инициализации некоторой части системы и после ее окончания, можно оценить объем динамической памяти, выделенной при инициализации этой части.

Узнать сколько динамической памяти выделено при начальной инициализации всех частей ОСРВ можно, если после окончания инициализации в пользовательской функции вызвать функцию `mem_showS()` и посмотреть значение, указывающее на размер выделенной памяти.

## Построение целевого образа, близкого к минимальному

Для разработки программного обеспечения под управлением ОСРВ Багет, используется комплекс, состоящий из инструментальной и целевой ЭВМ, которые соединены между собой. Инструментальная ЭВМ представляет собой персональный компьютер с процессором архитектуры Intel и операционной системой Linux. Вся разработка ведется на инструментальной ЭВМ с помощью соответствующих инструментальных средств, включающих в себя C-компилятор и удаленный отладчик ОСРВ Багет. Для создания компактной ОСРВ Багет в состав инструментальных средств включена еще утилита `mini`, описываемая в этой работе. Целевая ЭВМ с радиационно-стойким процессором является объектом для которого проводится работы по созданию компактной ОСРВ. Общий объем оперативной памяти определяется количеством банков памяти подключенных к микропроцессору (от 1 до 4). Размер каждого банка 4 Мбайта, таким образом размер памяти у процессора может варьироваться от 4 до 16 Мбайт. Такой небольшой размер оперативной памяти определяется требованиями надежности, помехозащищенности и сбоеустойчивости, для чего память оснащается развитой системой контроля и коррекции одиночных ошибок памяти. Кроме оперативной памяти целевая ЭВМ оснащена флэш-памятью размером 124 Мбайта. Описываемый отладочный стенд оснащен целевой ЭВМ с 4 Мбайтами оперативной памяти.

Создадим для этого целевой образ ОСРВ Багет, который будет иметь размер, близкий к минимальному, и состоять только из обязательных частей. Для чего при конфигурировании ОСРВ будем руководствоваться следующими соображениями:

- размер стека потоков управления можно задать равным 8196;

- исключить все, что можно не включать в образ и запретить использование плавающих чисел;

- максимально возможное количество мьютексов следует уменьшить до 10;

- в качестве пользовательской программы надо включить модуль “`mem_show.c`”, разработанный для тестирования образа, состоящего только из обязательных частей ОСРВ Багет.

При такой конфигурации получим образ по размерам близкий к минимальному. Для оценки статической памяти, требуемой образу, воспользуемся утилитой `mini` (таблица 8). Как видно из представленной таблицы минимальному образу требуется около 105 Кбайтов статической памяти. Далее представлена таблица 9, в которой содержится распределение памяти по отдельным модулям, входящих в состав минимального образа.

В качестве прикладной программы в минимальном образе используется модуль `mem_show.c`, который содержит следующие функции:

`void ti_list(int *arr)` – аргумент функции указывает на переменную типа `int`, в которую записывается количество потоков, выполняемых в системе;

`void mem_showS(int *arr)` – аргумент функции указывает на массив целых чисел, в который записываются три числа – сколько всего памяти содержится в системном пуле, сколько памяти уже выделено из системного пула, сколько памяти еще свободно в системном пуле.

`void *t_sleep(void *p)` – функция выполняет бесконечный цикл, внутри которого значение параметра цикла записывается по адресу, на который указывает переданный параметр, далее вызывается функция `sleep(1)`;

`int mem_show()` – стартовая функция модуля.

Адрес начала сохраняемой памяти обозначим, как `BEG_ADDR`. После загрузки созданного образа на целевую ЭВМ стартует функция `mem_show()`, которая выполняет следующие действия:

- вызывает функцию `mem_showS(BEG_ADDR + 0x20)`;

- порождает поток управления, внутри которого выполняется функция `t_sleep(BEG_ADDR + 0x2c)`;

- ожидает некоторое, чтобы созданный поток стартовал;

- вызывает функцию `mem_showS(BEG_ADDR + 0x30)`;

– вызывает функцию `ti_list(BEG_ADDR + 0x40)`;  
 – запускается цикл, который выполняется 100 раз, внутри цикла значение счетчика цикла записывается по адресу `BEG_ADDR + 0x3c` и выполняется функция `sleep(1)`, по окончании цикла производится перезагрузка.

После перезагрузки можно посмотреть содержимое названных адресов сохраняемой памяти:

```
di BEG_ADDR+0x20
BEG_ADDR+0x20 003e2b40 00004c44
003ddd8 0064
BEG_ADDR+0x30 003e2b40 00006e70
003dbcac 0063
BEG_ADDR+0x40 00000002 00000000
00000000 0000
```

Из представленной таблицы видно, что всего в системном пуле 3.88 Мбайтов памяти (`0x3e2b40`), до старта второго потока управления выделено 19 Кбайтов памяти (`0x4c44`) и 8 Кбайтов памяти было выделено на стек для второго потока управления (`0x6e70 – 0x4c44`). Содержимое адреса (`BEG_ADDR + 0x40`) подтверждает, что выполнялось два потока управления, а содержимое адресов (`BEG_ADDR + 0x2c`) и (`BEG_ADDR + 0x3c`) показывает количество выполненных циклов.

Этот пример позволяет оценить минимальное потребление памяти для многопоточной пользовательской задачи, выполняемое под управлением ОСРВ Багет.

Type	.text	.rodata	.data	.bss	.sdata	.sbss	TOTAL
os	32936	916	1096	5400	52	200	40600
init_os	14912	3392	300	4864			23468
mips	14508	2004	0	56	16	36	16620
PPM	7548		16	4		4	7572
mtx	5460	148	0	388	4	4	6004
mem	4316	92	0	984	16		5408
gcclib	3812	256	0	0			4068
string	752	512	0	0			1264
setjmp	612		0	0			612
user	560		0	0			560
stdlib	504		0	0			504
shell	0		0	0			0
<b>TOTAL</b>	<b>85920</b>	<b>7320</b>	<b>1412</b>	<b>11696</b>	<b>88</b>	<b>244</b>	<b>106680</b>

Таблица 8. Статическая память, требуемая образу, близкому к минимальному

Type	.text	.rodata	.data	.bss	.sdata	.sbss	TOTAL
=====os:							
hash.o	3632		0	160		4	3796
.	.	.	.	.	.	.	.
q_ddl.o	1308	32	800	0	16		2156
<b>total os</b>	<b>32936</b>	<b>916</b>	<b>1096</b>	<b>5400</b>	<b>52</b>	<b>200</b>	<b>40600</b>
=====init_os:							
oc2000.o	12808	3392	300	4864			21364
sscaling.o	2104	0	0			2104	
<b>total init_os</b>	<b>14912</b>	<b>3392</b>	<b>300</b>	<b>4864</b>			<b>23468</b>
.	.	.	.	.	.	.	.
=====string:							
s_memcpy.o	336		0	0			336
.	.	.	.	.	.	.	.
<b>total string</b>	<b>752</b>	<b>512</b>	<b>0</b>	<b>0</b>			<b>1264</b>
=====setjmp:							
sigsetjmp.o	612		0	0			612
=====user:							
mem_show.o	560		0	0			560
=====stdlib:							
malloc.o	504		0	0			504

Таблица 9. Распределение памяти по отдельным модулям, входящих в состав образа, близкого к минимальному

## Заключение

Разработаны средства и методы, позволяющие оценивать потребность целевого образа и его отдельных частей в статической памяти на инструментальной ЭВМ до его выполнения.

Определены точки использования специально разработанной функции для оценки потребностей в памяти, выделяемой динамически.

Создан отладочный стенд для разработки систем, в которых накладываются огра-

ничения на объем используемой оперативной памяти.

На инструментальной ЭВМ этого стенда добавлены средства, позволяющие контролировать потребность в оперативной памяти разрабатываемых систем.

Публикация выполнена в рамках государственного задания по проведению фундаментальных научных исследований (ГП14) по теме (проекту) «Исследование принципов построения компактной операционной системы для отечественных радиационно-стойких процессоров.» (№ 0065-2018-0021).

# Methods and tools for estimating the amount of memory required for an RTOS Baget image.

A.N.Godunov, V.A.Soldatov

**Abstract:** There is a class of hard real-time tasks for which RAM is a limited resource. The development of methods for reducing the amount of memory required for the RTOS implies the presence of a testing stand and tools for evaluating the amount of memory required to run the resulting image. All the memory required by the executable image can be divided into static and dynamic. Static memory is the memory required for the program code and image data sections. To estimate the amount of static memory required, it is proposed to use a specially designed utility that runs on an instrumental computer and allows to estimate how much static memory is required and how much memory is required for every part of the image. Dynamic memory refers to the memory that is allocated to the program at runtime. To estimate the required amount of dynamic memory, an appropriate function has been developed and points of its application are indicated. A testing stand designed specially for the development of a compact operating system is described.

**Keywords:** RTOS, scaling, configuration, static memory, dynamic memory, the ELF format, RTOS Baget

## Литература

1. А.А.Асонов, А.Н.Годунов, В.А.Солдатов. Методы снижения ресурсоемкости операционной системы жёсткого реального времени // Труды НИИСИ РАН. Том 8, №1, 2018 г.
2. В.Л.Безруков, А.Н.Годунов, П.Е.Назаров, В.А.Солдатов, И.Г.Хоменков. Введение в ос2000 // Вопросы кибернетики, М.: НИИСИ РАН, 1999 г., с. 76-106.
3. А.Н.Годунов, В.А.Солдатов. Операционные системы семейства Багет (сходства, отличия и перспективы) – «Программирование», Москва, 2014, №5, с. 68-76.
4. Е.А.Герлиц, В.В.Кулямин, А.В.Максимов, А.К.Петренко, А.В.Хорошилов, А.В.Цыварев. Тестирование операционных систем // Труды Института системного программирования РАН. Том 26, выпуск 1, 2014 г.

# О некоторых подходах к построению цифровых моделей 3D-объектов и формированию виртуальных коллекций

А.Н. Сотников<sup>1</sup>, И.Н. Соболевская<sup>2</sup>, И.Н. Чередниченко<sup>3</sup>

*МСЦ РАН, филиал ФГУ ФНЦ НИИСИ РАН, Москва, Россия,*

*E-mail's: <sup>1</sup>ASotnikov@jsc.ru, <sup>2</sup>nikfirst@jsc.ru, <sup>3</sup>inch@jsc.ru*

**Аннотация:** В данной статье рассмотрены некоторые подходы к построению цифровых моделей 3D-объектов и формированию виртуальных коллекций на основе полученных цифровых объектов. Рассмотрен, в частности, подход, основанный на методе фотограмметрии - трехмерной реконструкция объекта по стереотипам фотоизображениям. Описаны подходы, основанные на использовании технологий лазерного 3D-сканирования, а также, рассказано о подходе к построению цифровых 3D-объектов с использованием сканеров со структурированной подсветкой. Описано для каких наборов данных реконструкция 3D-модели методом фотограмметрии невозможна. Также рассмотрены виды технологий лазерного 3D-сканирования. Предложена технология представления виртуальных коллекций через интернет.

**Ключевые слова:** 3D-модель, фотограмметрия, лазерное сканирование, электронная библиотека, музейный объект, виртуальная выставка

## Введение

По мере развития и использования новых технологических решений, создаются инструменты, позволяющие перенести образы физических объектов в виртуальное пространство и внедрить виртуальный контент в физическое пространство, например, музеев.

Создание электронного музейного фонда преследует следующие основные цели:

- цифровая копия музейного объекта может быть использована для реставрационного обеспечения в случае утраты/повреждения объекта/ естественного старения артефакта;

- цифровая копия музейного объекта может быть использована для воссоздания предполагаемого первоначального образа объекта. Например, поврежденная часть скульптуры, найденной при археологических раскопках, может быть восстановлена с помощью специализированного программного обеспечения на основе цифрового образа исходного объекта;

- цифровая копия музейного объекта может быть использована для обеспечения доступности использования музейных предметов. В музейных фондах хранятся предметы, в количественном соотношении значительно превышающие возможности показа в музейной экспозиции. Для предоставления более полной научно-просветительской информации музей может обеспечить демонстрацию коллекционного фонда форме открытого доступа, т. е. в форме виртуальных коллекций.

Таким образом, формирование цифровых музейных фондов должно осуществляться в режиме обеспечения физической сохранности объекта, в частности, с использованием оборудования для бесконтактной оцифровки. Кроме того, особую роль в создании электронных музейных коллекций играет 3D-моделирование и формирование на основе цифровых моделей музейных 3D-объектов виртуального музейного пространства, как эффективного средства охраны, изучения и предоставления культурно-просветительского наследия.

Современное компьютерное оборудование и программное обеспечение позволяет применять 3D-моделирование во многих областях. Например, 3D-объекты широко применяются при проектировании деталей различной сложности, также трехмерная визуализация является одним из маркетинговых инструментов, интерактивных тренингов, презентаций.

Для достижения наиболее реалистичного эффекта трехмерную модель объекта можно «текстурировать», т.е. отразить визуальные особенности материала, задать освещение, анимировать.

Задачи трехмерного моделирования можно разделить на два принципиально различных класса:

- создание 3D-модели по заданным рисункам/чертежам, возможно, даже не существующего объекта;

- создание 3D-модели, аппроксимирующей поверхность «реально» существующего объекта.

В данной статье речь пойдет о подходах к формированию и построению цифровых 3D-объектов «физически» имеющих у «реконструктора». Т.е., предполагается, что мы имеем дело с самим объектом и наша задача сформировать и визуализировать 3D-модель этого объекта.

Современные методы формирования 3D-моделей основаны на построении полигональной сетки, аппроксимирующей реальную поверхность сканируемого объекта. Такая сетка может быть построена:

- программными средствами на основе обработки связанных пар стереоизображения объекта. Такой подход основан на методе фотограмметрии – трехмерной реконструкции объекта по стереотипам фотоизображениям;

- посредством применения специализированного оборудования – 3D-сканера, обрабатывающего дополнительную информацию о расстоянии до точек привязки, формируемых структурированной сеткой. В результате работы этих устройств полигональная сетка формируется динамически в процессе обработки искажений структурированной сеткой.

Такие сканеры бывают 2-х видов:

- лазерные сканеры, использующие прямое измерение расстояний до опорных точек;

- оптические сканеры, использующие внешнюю (структурированную) подсветку, формирующую структурную сетку и обрабатывающую информацию о ее искажениях, связанных с рельефом поверхности сканируемого объекта.

## 1.1. Фотограмметрия – трехмерная реконструкция объекта по фотоизображениям

Современные технологии создания трехмерных моделей высокого качества позволяют использовать фотографии, снятые цифровыми фотокамерами с матрицей разрешения менее 5 МПикс. А современные программы содержат процедуры калибровки объективов.

При этом съемка может производиться с любых ракурсов при условии, что каждый элемент реконструируемой сцены виден, по крайней мере, с двух позиций съемки.

Задача построения и визуализации 3D-поверхности методом фотограмметрии складывается на четыре этапа:

1. Определение параметров камер: положение, ориентацию, внутреннюю геометрию (фокусное расстояние, параметры дисторсии и т.п.). Результатом этого этапа будет

формирование набора позиций камеры. На основе этого набора выделяются особые точки, из которых формируется разреженное облако. А затем производится расчет геометрии с уплотнением.

2. Построение плотного облака точек, основанное на полученном на первом этапе разреженном облаке.

3. Построение трехмерной поверхности: полигональной модели и/или карты высот. Трехмерная полигональная модель описывает форму объекта, на основании плотного облака точек.

4. Построение текстуры для полигональной модели.

Пусть имеется набор снимков (рис. 1).



Рис. 1. Набор снимков для построения цифровой 3D-модели объекта программным средством, основанным на методе триангуляции.

Для построения цифровой 3D-модели объекта любым программным средством, основанным на методе триангуляции, на этих снимках осуществляется поиск одинаковых точек объекта для решения системы нелинейных уравнений, составленной на основе найденных соответствий.

Для этого необходимо задать фокусное расстояние камеры, положение, ориентацию камеры в моменты фотографирования каждого из снимков относительно, например, первого снимка (базового). Затем, выполняется поиск всех одинаковых точек на соседних парах изображений, после чего вычисляется положение точек в пространстве в системе координат базового снимка. Полученное, в результате решения системы нелинейных уравнений, множество точек представляется в виде триангуляционных каркасов, основанных на триангуляции Делоне [3], используется для текстурирования объекта, либо для трансформирования снимков. Для некоторых наборов данных реконструкция 3D-модели невозможна. Например, при съемке нетекстурированных, отражающих и прозрачных объектов (рис. 2). На рисунке 2 красная стрелка указывает на «маркер» (в данном случае каплю), позволяющий «глазу» увидеть смещение объекта. Однако, при обработке таких снимков программные средства, основанные на

описанном выше методе, «не видят» смещения камеры [4].

## 1.2. Лазерное 3D-сканирование, использующее прямое измерение расстояний до опорных точек

Принцип работы лазерного 3D-устройства заключается в плавном перемеще-

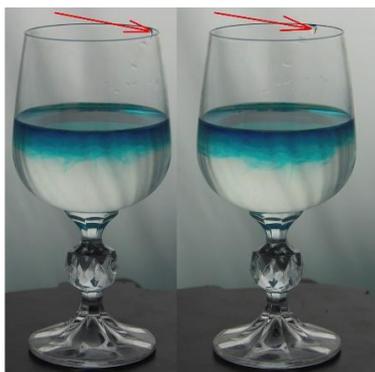


Рис. 2. Прозрачный объект.

нии измерительной головки сканера над сканируемым объектом, измерении расстояния в текущей точке головки сканера и объектом и записи получаемых координаты всех измеряемых точек. Тем самым формируется облако точек, которое, в свою очередь, после, пересчета и аппроксимации преобразуется в плотное облако точек, описанное выше, что значительно ускоряет процесс построения 3D-модели.

Лазерная технология, основанная на принципе работы лазерных дальномеров. Лазерная точка, камера и лазерный излучатель образуют «треугольник», в котором известны: расстояние между камерой и лазерным излучателем, расстояние между лазерным излучателем и лазерной точкой и угол падения лазерного луча на объект. Эти 3 показателя полностью определяют форму и размер треугольника. Чтобы ускорить процесс получения данных, вместо лазерной точки пользуются лазерной полосой. Облако точек создаётся с помощью триангуляции лазерной полосой.

## 1.3. Оптическое 3D-сканирование, использующее структурированную подсветку

Эта технология основана на принципе использования структурированного света. На сканируемый объект проецируется световая

сетка, деформация этой сетки и представляет собой модель сканируемого предмета. Сетка проецируется на объект с помощью жидкокристаллического проектора или другого постоянного источника света. Камера, расположенная чуть в стороне от проектора, фиксирует форму сети и вычисляет расстояние до каждой точки в поле ее зрения. Таким образом, в каждый момент времени сканируется несколько точек. Применение такой технологии позволяет получать точный результат за меньшее по сравнению с лазерной триангуляцией время. Преимущество 3D-сканеров, использующих структурированный свет, заключается в их скорости и точности работы.

Проведенные практические эксперименты позволили сравнить результаты 3D-моделирования, полученные описанными выше 3 способами. На основании проведенного анализа можно сделать следующие выводы:

- Применение ни одного из данных методов не дает возможности построить полноценную трехмерную модель объекта, имеющего прозрачную, светоотражающую или бликующую поверхность. Для построения 3D-моделей таких объектов, необходимо применение специального оборудования, например, лазера, создающего опорную сетку подсветки, и работающего в таком диапазоне длин волн, в котором «проблемные» поверхности сканируемого объекта непрозрачны. Возможно, что такой источник подсветки опорной сетки должен иметь настраиваемую рабочую частоту для работы с объектами из различных материалов.

Современные 3D-сканеры не позволяют получить полигональную сетку для построения 3D-модели «прозрачного» объекта без нанесения на его поверхность специального «маркирующего» средства, поглощающего свет. Очевидно, что такой способ сканирования невозможен для музейных объектов.

В свою очередь, применение метода фотограмметрии не требует применения никаких специальных средств, наносимых на поверхность фотографируемого объекта, но требует значительно больших по сложности и времени вычислений программных средств. Кроме того, построение полноценной 3D-модели методом фотограмметрии менее финансово затратный способ, по сравнению с использованием профессиональных сканеров со структурированной подсветкой.

## 2. Представление цифровых 3D-объектов

Виртуальная 3D-модель позволяет получить и сохранить большой объем информации

о пространственных характеристиках объекта, чем текстовое описание, рисунок, схема, фото- или кинодокумент. Особенно существенным это преимущество является для объектов со сложной геометрией или рельефом. Поэтому цифровые 3D-модели находят все более широкое применение, в том числе, в научно-технических задачах, в культурно-просветительских направлениях.

Цифровая 3D-модель представляет собой объект, который может быть связан посредством взаимных ссылок с другими типами электронных объектов: текстовыми, архивными, аудио- и видеодокументами и т.п. Эти ссылки хранятся, на удаленных серверах, доступ к которым может осуществляться через интернет. Такой мультимедийный объект, включающий в себя набор цифровых 3D-моделей, связанную с ними дополнительную информацию, может быть интегрирован в некое общее пространство знаний. Одним из эффективных средств интеграции и представления таких информационных объектов является электронная библиотека. Разработанная и используемая в электронной библиотеке «Научное наследие России» (ЭБ ННР), система описания цифровых образов музейных предметов и коллекций, обеспечивающая их интеграцию в единый информационный ресурс с возможностью эффективной навигации в нем, позволяет формировать тематические коллекции [5]. Кроме того, платформа ЭБ ННР позволяет создавать виртуальные выставки. Для создания виртуальной коллекции 3D-моделей с целью предоставления ее широкому кругу пользователей через интернет используется,

так называемая, технология интерактивной анимации. Эта технология основана на программной смене (прокручивании) фиксированного набора видов объекта (кадров) с помощью специализированных интерактивных программ отображения, имитирующих смену точки взгляда на исходный объект. Для создания такого интерактивного мультфильма, необходим набор заранее снятых сцен, которые будут использоваться как отдельные кадры экспозиции.

## Заключение

В настоящее время наблюдается определенный дефицит методов оцифровки музейных предметов, в первую очередь, для построения цифровых 3D моделей, их интеграции в единый тематический ресурс, а также предоставление широкому кругу пользователей средствами современных мультимедийных технологий. Поэтому, технология создания междисциплинарных тематических коллекций, основанная на фондах архивов, библиотек и музеев нуждается в разработке единого подхода к формированию таких коллекций и обеспечения доступа к ним. Однако, технология формирования виртуальных выставок, является принципиально новым эффективным решением задачи представления цифровых 3D-коллекций музейных предметов, интегрированных средствами электронной библиотеки. Работа выполнена в рамках Государственного задания. При проведении исследований использовался суперкомпьютер МВС-10П.

# About some methods of the construction of digital models of 3D-objects and the formation of virtual collections

A.N. Sotnikov, I.N. Sobolevskaya, I.N. Cherednichenko

**Annotation:** This article is about some approaches to the construction of digital models of 3D-objects and the formation of virtual collections based on the resulting digital objects. The method based on photogrammetry (a three-dimensional reconstruction of an object according to stereotypes of photographic images) is described too. The approaches based on the use of 3D laser scanning technology are described, as well as an approach to the construction of digital 3D objects using scanners with structured backlighting is described. Described for which data sets the reconstruction of the 3D-model by photogrammetry is impossible. Also considered are the types of 3D laser scanning technology. The technology of presentation of virtual collections via the Internet is proposed.

**Keywords:** 3D-model, photogrammetry, laser scanning, electronic library, museum object, virtual exhibition.

## Литература

1. И.К.Киямов, Р.Х.Мингазов, А.Ф.Музафаров, Р.А.Ибрагимов, А.А.Сибгатуллин. Технология лазерного сканирования в 3d проектировании // Экспозиция нефть газ. – 2013. № 7. – С. 41-43.
2. John P.Cooper, Andrew Wetherelt, Chiara Zazzaro. From Boatyard to Museum: 3D laser scanning and digital modelling of the Qatar Museums watercraft collection // International journal of nautical archaeology. V. 47, Iss. 2. – 2018. P. 419-442.
3. Yidan Yao, Chengda Lin, Ruifang Zhai. Rapeseed 3D Reconstruction and Morphological Parameter Measurement Based on Laser Point Cloud // International Conference on Agro-Geoinformatics / 2016. – P. 187-192.
4. Ю.Г.Бульчев, Е.Н.Чепель. Квазиоптимальный метод решения задачи триангуляции в условиях априорной неопределённости // Автометрия. Том 53. – 2017. № 6. – С. 83-91.
5. Н.Е. Каленов, И.Н.Соболевская, А.Н.Сотников. Цифровые музейные коллекции и представление объектов естественнонаучного музейного хранения в электронной библиотеке «Научное наследие России». // НТИ СЕР. 1.- 2016. № 10.- С. 33-38.
6. М.А.Сафрошкин. Алгоритмы обработки изображений для систем регистрации трехмерных объектов, использующих структурированную подсветку с цветовым кодированием. // Молодежный научно-технический вестник ФС77-51038, ISSN 2307-0609 - <http://sntbul.bmstu.ru/doc/749386.html>

# Аппаратная реализация алгоритма Adler32 для вычисления контрольных сумм

А.В.Крыницкий

ФГУ ФНЦ НИИСИ РАН, Москва, Россия, E-mail: alexis@cs.niisi.ras.ru

**Аннотация:** В статье рассказывается о вариантах аппаратной реализации алгоритма Adler32 для вычисления контрольных сумм и описывается разработка синтезируемых моделей на языке Verilog.

**Ключевые слова:** Adler32, аппаратная реализация, Verilog.

## I. Введение

В процессе проектирования контроллера Gigabit Ethernet возникла необходимость в блоке аппаратного вычисления контрольной суммы пакета по алгоритму Adler32. Этот алгоритм применяется в протоколе транспортного уровня SCTP (Stream Control Transmission Protocol, RFC2960).

Контрольная сумма Adler32 получается путём вычисления двух 16-разрядных контрольных сумм А и Б и объединении их в 32-разрядное целое. Сумма А получается сложением всех байт в пакете с последующим делением по модулю 65521. Сумма Б получается сложением всех отдельных значений А на каждом шаге с последующим делением по модулю 65521. В начале А инициализируется единицей, а Б нулем. Так, для пакета, состоящего из байтов  $D_1, D_2, \dots, D_n$  ( $n$  - количество байт в пакете), можно записать:

$$A = [1 + D_1 + D_2 + \dots + D_n] \pmod{65521} \quad (1)$$

$$B = [(1 + D_1) + (1 + D_1 + D_2) + \dots + (1 + D_1 + D_2 + \dots + D_n)] \pmod{65521}$$

$$\begin{aligned} & \text{или} \\ B &= [n + n \times D_1 + (n-1) \times D_2 + \\ & (n-2) \times D_3 + \dots + D_n] \pmod{65521} \end{aligned} \quad (2)$$

$$\text{Adler32} = \{B, A\} = B \times 65536 + A \quad (3)$$

Источники [1] и [2] содержат программные алгоритмы расчета Adler32 и не применимы для аппаратной реализации. Но некоторые идеи, выкладки и формулы, приведённые в них, были использованы в данной работе.

## II. Постановка задачи

Все внутренние шины в контроллере Gigabit Ethernet имеют ширину 8 байт и работают на тактовых частотах до 125МГц.

Соответственно, разрабатываемый блок Adler32 должен обрабатывать 8 байт за один период тактового сигнала 125МГц (а лучше, если не менее 157 МГц - с прицелом на будущий контроллер 10Gigabit). Кроме того, как минимум, первое и последнее 8-байтовое слово могут не содержать все 8 байт данных, т.е. блок должен уметь работать с не выровненными словами.

Результатом разработки должна стать корректно работающая, синтезируемая RTL-модель на языке Verilog. Не ставилась цель разработать схему с максимально возможным быстродействием или минимальным числом используемых ресурсов (логических вентилях, триггеров и т.д.). Также не ставилась задача разработки универсальной, параметризуемой модели. Критерием удовлетворения частотным требованиям является отсутствие временных нарушений при разводке проекта блока в микросхему FPGA Altera EP4SGX530NF45C3 - наиболее широко используемую в НИИСИ РАН микросхему для FPGA-прототипирования.

## III. Теория

### 1. Суммирование

Пусть  $a$  - искомая сумма А,  $a[0]$  - некое начальное значение суммы А,  $b$  - искомая сумма Б,  $b[0]$  - начальное значение суммы Б,  $c[i]$  - байт данных с номером  $i$  (байты нумеруются от 0 до  $n-1$ ). Тогда значение суммы А, после обработки первого байта данных:

$$a[1] = a[0] + c[0].$$

Значение суммы Б, после обработки первого байта данных:

$$b[1] = b[0] + a[1] = b[0] + a[0] + c[0].$$

Значение суммы А, после обработки второго байта данных:

$$a[2] = a[1] + c[1] = a[0] + c[0] + c[1].$$

Значение суммы Б, после обработки второго байта данных:

$$\begin{aligned} b[2] &= b[1] + a[2] = \\ b[0] + a[0] + c[0] + a[0] + c[0] + c[1] &= \\ b[0] + 2 \times a[0] + 2 \times c[0] + c[1]. \end{aligned}$$

Значение суммы А, после обработки последнего байта данных:

$$a[n] = a[n-1] + c[n-1] = a[0] + \sum_{i=0}^{n-1} c[i]. \quad (4)$$

Значение суммы Б, после обработки последнего байта данных:

$$\begin{aligned} b[n] &= b[n-1] + a[n] = \\ b[0] + n \times a[0] + \sum_{i=0}^{n-1} ((n-i) \times c[i]). \end{aligned} \quad (5)$$

Таким образом, для 8-байтового слова данных справедливо:

$$a[8] = a\_reg + \sum_{i=0}^7 c[i], \quad (6)$$

где *a\_reg* – регистр, содержащий предыдущее значение суммы А;

$$b[8] = b\_reg + 8 \times a\_reg + \sum_{i=0}^7 ((8-i) \times c[i]), \quad (7)$$

где *b\_reg* – регистр, содержащий предыдущее значение суммы Б.

Разворачивая формулы (6) и (7) в строку, получаем:

$$\begin{aligned} a[8] &= a\_reg + c[0] + c[1] + c[2] + \\ c[3] + c[4] + c[5] + c[6] + c[7]; \end{aligned} \quad (8)$$

$$\begin{aligned} b[8] &= b\_reg + 8 \times a\_reg + 8 \times c[0] + \\ 7 \times c[1] + 6 \times c[2] + 5 \times c[3] + 4 \times c[4] + \\ 3 \times c[5] + 2 \times c[6] + c[7]. \end{aligned} \quad (9)$$

## 2. Деление по модулю 65521

Если для двух чисел М и N справедливо равенство  $M = K \times N + R$  (K и R – целые неотрицательные числа,  $R < N$ ), то результатом деления числа М по модулю N является число R:  $M \bmod N = R$ .

У деления по модулю имеется 2 полезных свойства:

$$\begin{aligned} (M + N) \bmod G &= \\ ((M \bmod G) + (N \bmod G)) \bmod G, \end{aligned} \quad (10)$$

$$\begin{aligned} (M \times N) \bmod G &= \\ ((M \bmod G) \times (N \bmod G)) \bmod G. \end{aligned} \quad (11)$$

Операция деления в Verilog, вообще говоря, не является синтезируемой конструкцией. Существуют различные методы обойти это ограничение. В данной работе деление по модулю 65521 было

решено делать следующим методом: делимое последовательно сравнивается с числом 65521, умноженным на целочисленный коэффициент, начинающийся с 1 и инкрементирующийся на каждом новом шаге сравнения; на каком-то шаге делимое будет меньше; за результат деления принимается разность делимого и произведения 65521 на коэффициент из предыдущего шага. Фрагмент verilog-кода схемы деления 19-разрядного входного числа по модулю 65521 приведён на рисунке 1.

```
function [15:0] MOD19;
input [18:0] in;
reg m_done;
begin
m_done = 0;
MOD19 = 16'd0;
for(k=1;k<=9;k=k+1) begin
if(m_done==0) begin
if(in<k*16'd65521) begin
m_done=1;
MOD19 = in - (k-1)*16'd65521;
end end end end
endfunction
```

Рисунок 1. Фрагмент verilog-кода схемы деления 19-разрядного входного числа по модулю 65521

Число повторений цикла сравнения (в примере равно 9) вычисляется заранее из следующих соображений:

$2^{19} - 1 = 524287 = 8 \times 65521 + 119$ , (12)  
поэтому число повторений  $8 + 1 = 9$ . Общая же формула для вычисления количества повторений K для N-разрядного числа такая:

$$\frac{(2^N - 1)}{65521} = (K - 1) + r, (r < 1). \quad (13)$$

## IV. Реализация

### 1. Первый этап

На первом этапе модель реализовывала алгоритм Adler32 самым очевидным и прямым способом, в строгом соответствии с формулами (8) и (9). Сложение сумм А и Б и деление по модулю осуществлялись за один период тактового сигнала с «защёлкиванием» полученных результатов для использования со следующим словом данных. Для поддержки не выровненных слов был введён входной вектор ВЕ, каждый разряд которого явно указывал действителен ли байт или нет. Для упрощения процесса

проектирования было наложено ограничение на формат входного слова: все действительные байты должны быть выровнены к нулевому байту, т.е. разрешёнными значениями для ВЕ являются значения 0x1 для слова, содержащего 1 байт (байт 0), 0x3 - для слова, содержащего 2 байта (байты 1-0), ..., 0x7f - для слова, содержащего 7 байт (байты 6-0), 0xff - для слова, содержащего все 8 байт.

Фрагмент verilog-кода для расчёта сумм А и Б приведён на рисунке 2.

```
reg next_done;
reg [19:0] sum_a_tmp;
reg [19:0] sum_b_tmp;
always @(*) begin
    sum_a_tmp = sum_a_reg;
    sum_b_tmp = sum_b_reg;
    next_done = 0;
    for(i=0; i<8; i=i+1) begin
        if(next_done==0) begin
            if(!b[i]) next_done=1;
        else begin
            sum_a_tmp = sum_a_tmp + byte[i];
            sum_b_tmp = sum_a_tmp + sum_b_tmp;
        end end end
    sum_a = MOD20(sum_a_tmp);
    sum_b = MOD20(sum_b_tmp);
end
```

Рисунок 2. Фрагмент verilog-кода первой схемы расчёта контрольной суммы по алгоритму Adler32 (расчёт сумм А и Б)

Фрагмент verilog-кода для расчёта финальной суммы приведён на рисунке 3.

```
always @(posedge clk or negedge reset)
if(!reset) begin
    sum_a_reg <= 16'd1;
    sum_b_reg <= 16'd0;
end
else if(!init) begin
    sum_a_reg <= 16'd1;
    sum_b_reg <= 16'd0;
end
else if(!enable) begin
    sum_a_reg <= sum_a;
    sum_b_reg <= sum_b;
end
wire [31:0] final_csum = {sum_b_reg, sum_a_reg};
```

Рисунок 3. Фрагмент verilog-кода первой схемы

расчёта контрольной суммы по алгоритму Adler32 (финальная сумма)

Выбор разрядности 20 для переменных  $sum\_a\_tmp$  и  $sum\_b\_tmp$  обусловлен максимальным значением, которое может принимать  $sum\_b\_tmp$ . Из формулы (5) следует, что, если взять максимальное начальное значение сумм А и Б (65520 каждая) и все байты N-байтного блока имеют значение 255 (0xFF), то получится, что

$$b\_max = b[0] + N \times a[0] + \sum_{i=0}^{N-1} ((N-i) \times c[i]) = 65520 \times (N+1) + N \times (N+1) \times \frac{255}{2}. \quad (14)$$

Для  $N=8$  получаем число 598860 = 0x9234C, имеющее 20 разрядов.

Функция MOD20 совпадает с функцией на рисунке 1. Только вместо значения  $k=9$  используется  $k=17$ , полученное с помощью формулы (13) для 20-разрядного слова.

Корректность функционирования модели проверялась с помощью образцов реальных SCTP-пакетов, взятых из сети Интернет. Модель обладает следующими достоинствами:

- простота,
- неограниченное количество обрабатываемых слов,
- любое слово может не содержать все 8 байт,
- отсутствие задержки выдачи результата (т.е. результат известен в следующем такте после «защёлкивания» последнего слова),
- относительно небольшое количество потребляемых ресурсов (1759 ALUTs / 37 registers).

Главным недостатком модели является слишком низкая частота работы – всего 61 МГц.

В дальнейшем эта модель использовалась в качестве поведенческой, как референсный источник контрольных сумм пакетов произвольной длины и произвольного содержания.

## 2. Второй этап

Анализ критических цепей схемы первого этапа показал, что наибольшую задержку вносят суммирование 8 байтов для расчёта суммы А и путь от регистра  $sum\_a\_reg$  до  $sum\_b\_reg$ . Поэтому, на втором этапе было решено распараллелить процессы сложения, разделить суммирование и деление по модулю и добавить промежуточные регистры. После рассмотрения несколько

вариантов было решено остановиться на таком:

1. Сумма А разбивается на 2 части: сумма байтов 3:0 и сумма байтов 7:4. Они считаются параллельно и результаты «защёлкиваются» в промежуточные регистры. Когда будет обработано последнее слово, результаты обоих сумм складываются, формируя финальную сумму А. И уже над ней затем производится деление по модулю.

2. Сумма Б также разбивается на 2 части: сумма промежуточных полусумм А из п.1, умноженная на количество байт в слове и сумма байтов 7:0, умноженных на соответствующие коэффициенты (формула 9). Когда будет обработано последнее слово, результаты обоих сумм складываются, формируя финальную сумму Б. И уже над ней затем производится деление по модулю.

## 2.1 Расчёт суммы А

Фрагмент verilog-кода, иллюстрирующий расчёт суммы А, приведён на рисунках 4 и 5.

```
reg [20:0] sumA_l; // halfsumA, bytes 3-0
reg [20:0] sumA_h; // halfsumA, bytes 7-4
always @(posedge clk or negedge reset)
if(!reset) begin
    sumA_l <= {{20{1'b0}},1'b1};
    sumA_h <= {{20{1'b0}},1'b0};
end
else if(!init) begin
    sumA_l <= {{20{1'b0}},1'b1};
    sumA_h <= {{20{1'b0}},1'b0};
end
else begin
if(enable) case(be)
8'b0000_0001: sumA_l <= sumA_l+byte[0];
8'b0000_0011: sumA_l <= sumA_l+byte[0]+
    byte[1];
8'b0000_0111: sumA_l <= sumA_l+byte[0]+
    byte[1]+byte[2];
    default: sumA_l <= sumA_l+byte[0]+
    byte[1]+byte[2]+byte[3];
endcase
```

Рисунок 4. Фрагмент verilog-кода распараллеленного расчёта контрольной суммы А (первая полусумма)

```
if(enable) case(be)
8'b0000_0001,
8'b0000_0011,
8'b0000_0111,
8'b0000_1111: sumA_h <= sumA_h;
8'b0001_1111: sumA_h <= sumA_h+
    byte[4];
8'b0011_1111: sumA_h <= sumA_h+
    byte[4]+byte[5];
8'b0111_1111: sumA_h <= sumA_h+
    byte[4]+ byte[5]+byte[6];
    default: sumA_h <= sumA_h+
    byte[4]+byte[5]+byte[6]+byte[7];
endcase
end
```

// sum A before modulo division

```
wire [21:0] sumA = sumA_l + sumA_h;
```

Рисунок 5. Фрагмент verilog-кода распараллеленного расчёта контрольной суммы А (первая полусумма)

На рисунках 4 и 5 *enable* - это сигнал разрешения (строб) очередного слова.

Регистры *sumA\_l* и *sumA\_h* аккумулируют результаты сложения полуслов и их разрядность определяется количеством обработанных байт. Следовательно, придётся отказаться от концепции неограниченного количества байт как в схеме №1 и ограничить длину пакета. Блок предназначен для использования в контроллере Gigabit Ethernet, а в сети Ethernet не должно быть пакетов длиннее 15.5кБайт (максимальная длина Jumbo-пакета). Поэтому было введено ограничение в 16384 байта. Отсюда максимальное значение в регистрах *sumA\_l* и *sumA\_h*:

$$\frac{16384}{2} \times 255 = 2088960 = 0x1FE000.$$

Это 21-разрядное число. Финальная сумма *sumA*, имеет, соответственно, 22 разряда. Деление по модулю 22-разрядного числа в соответствии с формулой (13) требует  $k=65$ . Это практически не реализуемо из-за огромной временной задержки и количества задействованной логики. Поэтому придётся разбить процесс деления на 2 этапа и задействовать 2 периода тактового сигнала. Для этого воспользуемся свойствами деления по модулю (10) и (11).

Представим число *sumA[21:0]* как сумму числа *sumA[21:19]*, сдвинутого на 19 разрядов влево, и 19-разрядного числа *sumA[18:0]*:

$$\begin{aligned} sumA[21:0] &= \\ sumA[21:19] \times 2^{19} &+ sumA[18:0]. \end{aligned}$$

Разрядность 19 подобрана эмпирически как максимальная, при которой ещё возможно достижение требуемых временных характеристик проекта для выбранной микросхемы FPGA.

$$\begin{aligned} (sumA[21:0]) \bmod 65521 &= \\ \left( \begin{aligned} &(sumA[21:19] \times 2^{19}) \bmod 65521 + \\ &(sumA[18:0]) \bmod 65521 \end{aligned} \right) \bmod 65521 &= \\ \left( \begin{aligned} &\left( (sumA[21:19]) \bmod 65521 \times \right. \\ &\left. (2^{19}) \bmod 65521 \right) \bmod 65521 + \\ &(sumA[18:0]) \bmod 65521 \end{aligned} \right) \bmod 65521 &= \\ \left( \begin{aligned} &sumA[21:19] * 120 + \\ &(sumA[18:0]) \bmod 65521 \end{aligned} \right) \bmod 65521. \end{aligned}$$

Фрагмент кода, реализующий вышесказанное, приведён на рисунках 6 и 7.

```
//mod(2^19)=mod(524288)=120
//mod(sumA[21:19])=sumA[21:19]
```

```
reg [9:0] modA_h;
reg [15:0] modA_l;
always @(posedge clk or negedge reset)
if(!reset) begin
    modA_h <= 10'd0; modA_l <= 16'd0;
end
else if(init) begin
    modA_h <= 10'd0; modA_l <= 16'd0;
end
else begin
    if(enable_d) modA_h <=
sumA[21:19]*7'd120;
    if(enable_d) modA_l <=
MOD19(sumA[18:0]);
end
```

Рисунок 6. Фрагмент verilog-кода поэтапного деления по модулю суммы А (первый этап)

```
reg [15:0] sumA_fin; // final sum A
always @(posedge clk or negedge reset)
if(!reset) sumA_fin <= 16'd1;
else if(init) sumA_fin <= 16'd1;
else if(enable_dd) sumA_fin <=
    MOD19(modA_h + modA_l);
end
```

Рисунок 7. Фрагмент verilog-кода поэтапного деления по модулю суммы А (второй этап)

На рисунках 6 и 7 сигналы *enable\_d* и *enable\_dd* – это задержанные, соответственно, на 1 и 2 такта копии сигнала разрешения очередного слова (*enable*). Таким образом, результат расчёта контрольной суммы А доступен на третий такт после обработки последнего слова.

## 2.2 Расчёт суммы Б

Фрагмент verilog-кода, иллюстрирующий расчёт суммы Б, приведён на рисунках 8 и 9.

В соответствии с формулой (13) максимальное значение суммы Б для  $N=16384$  равно  $34227609600 = 0x7F81FE000$ . Это 35-разрядное число. Такую разрядность и имеет переменная *sumB*, содержащая финальную сумму Б (до деления по модулю). Обе полусуммы, *sumB\_l* и *sumB\_h*, также объявлены как 35-разрядные.

```
always @(posedge clk or negedge reset)
```

```
if(!reset) begin
```

```
    sumB_l <= {{34{1'b0}},1'b0};
```

```
    sumB_h <= {{34{1'b0}},1'b0};
```

```
end
```

```
else if(init) begin
```

```
    sumB_l <= {{34{1'b0}},1'b0};
```

```
    sumB_h <= {{34{1'b0}},1'b0};
```

```
end
```

```
else begin
```

```
if(enable) case(be)
```

```
8'b0000_0001: sumB_l <= sumB_l +
    1*(sumA_l + sumA_h);
```

```
8'b0000_0011: sumB_l <= sumB_l +
    2*(sumA_l + sumA_h);
```

```
8'b0000_0111: sumB_l <= sumB_l +
    3*(sumA_l + sumA_h);
```

```
8'b0000_1111: sumB_l <= sumB_l +
    4*(sumA_l + sumA_h);
```

```
8'b0001_1111: sumB_l <= sumB_l +
    5*(sumA_l + sumA_h);
```

```
8'b0011_1111: sumB_l <= sumB_l +
    6*(sumA_l + sumA_h);
```

```
8'b0111_1111: sumB_l <= sumB_l +
    7*(sumA_l + sumA_h);
```

```
default: sumB_l <= sumB_l +
    8*(sumA_l + sumA_h);
```

```
endcase
```

Рисунок 8. Фрагмент verilog-кода распараллеленного расчёта контрольной суммы Б (первая полусумма)

```

if(enable) case(be)
8'b0000_0001: sumB_h <= sumB_h+
                byte[0];
8'b0000_0011: sumB_h <= sumB_h+
                2*byte[0]+byte[1];
8'b0000_0111: sumB_h <= sumB_h+
                3*byte[0]+2*byte[1]+byte[2];
8'b0000_1111: sumB_h <= sumB_h+
                4*byte[0]+3*byte[1]+2*byte[2]+
                byte[3];
8'b0001_1111: sumB_h <= sumB_h+
                5*byte[0]+4*byte[1]+3*byte[2]+
                2*byte[3]+byte[4];
8'b0011_1111: sumB_h <= sumB_h+
                6*byte[0]+5*byte[1]+4*byte[2]+
                3*byte[3]+2*byte[4]+byte[5];
8'b0111_1111: sumB_h <= sumB_h+
                7*byte[0]+6*byte[1]+5*byte[2]+
                4*byte[3]+3*byte[4]+2*byte[5]+
                byte[6];
default: sumB_h <= sumB_h+
                8*byte[0]+7*byte[1]+6*byte[2]+
                5*byte[3]+4*byte[4]+3*byte[5]+
                2*byte[6]+byte[7];

```

endcase

// sum B before modulo division

```
wire [34:0] sumB = sumB_1 + sumB_h;
```

Рисунок 9. Фрагмент verilog-кода распараллеленного расчёта контрольной суммы Б (вторая полусумма)

Деление по модулю 35-разрядного числа также будем осуществлять в несколько этапов.

```
sumB[34:0] =
sumB[34:19] × 219 + sumB[18:0];
```

$$\begin{aligned}
 & (sumB[34:0]) \bmod 65521 = \\
 & \left( \begin{array}{l} (sumB[34:19] \times 2^{19}) \bmod 65521 + \\ (sumB[18:0]) \bmod 65521 \end{array} \right) \bmod 65521 = \\
 & \left( \begin{array}{l} ((sumB[34:19]) \bmod 65521 \times \\ 120 \\ (sumB[18:0]) \bmod 65521 \end{array} \right) \bmod 65521
 \end{aligned}$$

На рисунках 10, 11 и 12 приведён фрагмент кода, иллюстрирующий процесс деления по модулю суммы Б. Сигналы *enable\_d*,

*enable\_dd* и *enable\_ddd* – это задержанные, соответственно, на 1, 2 и 3 такта копии сигнала разрешения очередного слова (*enable*).

```

reg [22:0] modB_h;
reg [15:0] modB_1;
always @(posedge clk or negedge reset)
if(!reset) begin
modB_h <= 23'd0;
modB_1 <= 16'd0;
end
else if(init) begin
modB_h <= 23'd0;
modB_1 <= 16'd0;
end
else if(enable_d) begin
modB_h <= MOD16(sumB[34:19])*7'd120;
modB_1 <= MOD19(sumB[18:0]);
end

```

Рисунок 10. Фрагмент verilog-кода поэтапного деления по модулю суммы Б (первый этап)

Переменная *modB\_h* имеет разрядность 23 и требует дальнейшего разбиения:

$$\begin{aligned}
 & (modB_h[22:0]) \bmod 65521 = \\
 & \left( \begin{array}{l} (modB_h[22:16] \times 2^{16}) \bmod 65521 + \\ (modB_h[15:0]) \bmod 65521 \end{array} \right) \bmod 65521 = \\
 & \left( \begin{array}{l} ((modB_h[22:16]) \times 15) + \\ (sumB_h[15:0]) \bmod 65521 \end{array} \right) \bmod 65521
 \end{aligned}$$

Это требует ещё одного уровня триггеров и, соответственно, теряется ещё один такт. После вычисления получается 16-разрядная переменная *modB\_h\_mult*.

```

reg [15:0] modB_h_mult;
always @(posedge clk or negedge reset)
if(!reset) begin
modB_h_mult <= 16'd0;
end
else if(init) begin
modB_h_mult <= 16'd0;
end
else if(enable_dd) begin
modB_h_mult <=
MOD19(modB_h[22:16]*15 +
MOD16(modB_h[15:0]));
end

```

end

Рисунок 11. Фрагмент verilog-кода поэтапного деления по модулю суммы Б (второй этап)

```
reg[15:0] sumB_fin;// final sum B
always @(posedge clk or negedge reset)
if(!reset) begin
sumB_fin <= 16'd0;
end
else if(!init) begin
sumB_fin <= 16'd0;
end
else begin
if(enable_ddd) sumB_fin <=
MOD19(modB_h_mult + modB_l);
end
////////////////////////////////////
// GRAND FINAL
wire [31:0] final_csum =
{sumB_fin,sumA_fin};
```

Рисунок 12. Фрагмент verilog-кода поэтапного деления по модулю суммы Б (третий этап)

Таким образом, результат расчёта контрольной суммы Б доступен лишь на четвёртый такт после обработки последнего слова. Зато получена частота 165МГц, т.е. цель достигнута.

## V. Заключение

Разработана синтезируемая модель аппаратного расчёта контрольной суммы по алгоритму Adler32.

Корректность функционирования модели проверялась с помощью образцов реальных SCTP-пакетов, взятых из сети Интернет, и поведенческой модели, полученной на этапе 1. С помощью неё генерировались контрольные суммы пакетов различной длины (в том числе максимальной - 16384 байта) и различного содержимого (в том числе целиком заполненные байтами со значением 0xFF).

Модель обладает следующими свойствами:

- 8-байтовые слова для параллельной обработки,
- любое слово может не содержать все 8 байт,
- максимально 16384 байт в пакете.

Плюсы:

- достаточно высокая частота – 165МГц.

Минусы:

- задержка выдачи результата 4 такта,
- довольно большое количество потребляемых ресурсов (2601 ALUTs / 231 registers).

# Hardware realization of Adler32 checksum algorithm

A.Krynitskiy

**Abstract:** This paper observes variants of hardware realization of the Adler32 checksum algorithm and describes synthesizable Verilog RTL model.

**Keywords:** Adler32, hardware realization, Verilog.

## Литература

- [1] Konstantinos Margaritis. Vectorization of Algorithm Adler32 Using Altivec // Nafplion, Greece, May 15,2005
- [2] James Guilford. Fast Computation of Adler32 Checksums // Intel Corporation, June 8, 2017.  
URL: <https://software.intel.com/en-us/articles/fast-computation-of-adler32-checksums>

# Модель расчета отраженного света для имитации на GPU физически корректного освещения объектов виртуальной среды

А.В. Мальцев<sup>1</sup>, Д.В. Омельченко<sup>2</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия, E-mail's: <sup>1</sup>avmaltcev@mail.ru, <sup>2</sup>omelchenko\_dv@mail.ru

**Аннотация:** В работе предлагается оригинальная вычислительная модель для распределенного расчета интенсивности света, отраженного от поверхности объектов трехмерной виртуальной среды. Предлагаемое решение является развитием модели Кука-Торренса, которое позволяет реализовывать в виртуальной сцене различные типы источников света и тени от объектов. Созданная модель обеспечивает возможность распределенного рендеринга виртуальной среды с имитацией физически корректного освещения на многоядерных графических процессорах в реальном времени.

**Ключевые слова:** визуализация, материал, распределенные вычисления, модель освещения, графический процессор.

## Введение

Важной составляющей процесса рендеринга трехмерной виртуальной среды является моделирование отражения света от поверхностей ее объектов.

При решении данной задачи применяются различные вычислительные модели, каждая из которых имеет свои плюсы и минусы.

Многие графические приложения построены на упрощенных моделях расчета [1, 2] (например, Фонга, Блинна и т.п.), поскольку они хорошо программируются на современных графических картах и обеспечивают визуализацию трехмерных виртуальных сцен в масштабе реального времени.

Однако их существенным минусом является недостаточный уровень реалистичности визуальных свойств моделируемых материалов, например, металлов.

Проблему реалистичности можно решить путем использования глобальных моделей освещения [3] (с использованием технологий фотонных карт, radiosity и др.), которые позволяют достичь качества синтезируемых изображений, сравнимого с фотоснимками.

Но, к сожалению, трудоемкость основанных на таких моделях методов и алгоритмов весьма высока, что не позволяет достичь реального времени при визуализации сложных виртуальных сцен даже с помощью современной вычислительной техники.

Поэтому использование глобального освещения в таких областях, как системы виртуального окружения и имитационно-тренажерные комплексы весьма затруднительно. Перспективным направлением в области виртуального моделирования и визуализации является создание новых эффективных методов и алгоритмов расчета освещенности поверхностей объектов на основе технологий PBR (*physically based rendering*).

Применение такого подхода при рендеринге трехмерных виртуальных сцен обеспечивает имитацию физически корректного освещения, при которой соблюдается баланс между качеством изображения и скоростью синтеза кадра. Решения, основанные на PBR-технологиях, используются в таких платных западных продуктах, как Unity и Unreal Engine 4.

Однако детали применяемых в них методов и алгоритмов, как правило, являются коммерческой тайной и не публикуются в открытых источниках.

В данной работе предлагается оригинальная модель вычисления отраженного света в точке поверхности виртуального объекта, в основу которой положены принципы PBR и BRDF модель Кука-Торренса [4].

Разработанная модель обеспечивает распределенную визуализацию сложных виртуальных сцен на современных

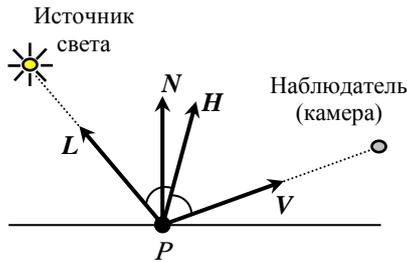


Рис. 1. Векторы для расчета освещения

графических процессорах (GPU) в масштабе реального времени.

## 1. Модель Кука-Торренса

Модель расчета освещения, предложенная Робертом Куком и Кеннетом Торренсом, базируется на представлении поверхностей виртуальных объектов в виде совокупности идеально отражающих плоских микрограней, имеющих различные ориентации. Данная модель учитывает рассеянную, диффузную и зеркальную составляющие отражения. Первая возникает в результате попадания на поверхность объекта и последующего равномерного отражения во всех направлениях того света, который был отражен от других объектов виртуальной среды. Две других связаны с отражением и преломлением световых лучей, генерируемых каким-либо источником и падающих на поверхность объекта. Так, диффузное отражение возникает в следствие многочисленных переотражений преломленных лучей внутри материала объекта и выхода их вновь в окружающую среду вблизи точек преломления. Зеркальная составляющая представляет собой свет, отраженный от поверхности согласно закону отражения.

Интенсивности рассеянного и диффузно отраженного света не зависит от положения наблюдателя. Рассеянная составляющая определяется, исходя из способности материала к отражению рассеянного света, которая задается дизайнером при создании материала в виде вещественных коэффициентов для  $R$ ,  $G$  и  $B$  компонент. Диффузная – вычисляется по закону Ламберта [5]. Основу модели Кука-Торренса составляет расчет зеркальной составляющей  $R_s$  отражения света от поверхности объекта в некоторой ее точке  $P$ . Для этого применяется следующая формула:

$$R_s = \frac{D \cdot G \cdot F}{\pi (V, N)(L, N)}, \quad (1)$$

где  $L$  – единичный вектор из  $P$  на источник света,  $V$  – единичный вектор из  $P$  на

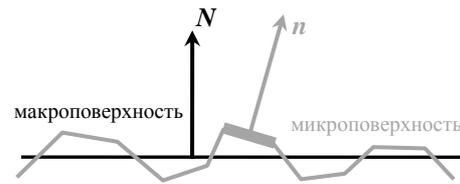


Рис. 2. Представление поверхности в виде микрограней

наблюдателя,  $N$  – нормаль к поверхности в рассматриваемой точке (рис. 1). В числителе располагается произведение трех функций, отвечающих соответственно за распределение наклона микрограней поверхности, моделирование самозатенения микрограней и расчет коэффициента Френеля. Рассмотрим эти функции подробнее.

### 1.1. Распределение наклона микрограней

Как уже было упомянуто выше, в рассматриваемой вычислительной модели Кука-Торренса считается, что поверхность любого виртуального объекта состоит из множества маленьких отражающих граней. Каждая такая грань имеет свою нормаль  $n$  (рис. 2), определяющую ее наклон относительно макроповерхности со средней нормалью  $N$  и направление отражения падающего света. Чем больше углы между векторами  $n$  и вектором  $N$ , тем более шероховатой является поверхность объекта, что приводит к рассеиванию отраженных лучей и повышению степени размытости зеркального отражения. Наклон микрограней регулируется в формуле (1) с помощью функции  $D$ , в качестве которой могут выступать различные функции распределения. Однако согласно работе [4], высокую степень сходства экспериментальных и моделируемых результатов обеспечивает применение распределения Бэкмана

$$D = \frac{e^{((H, N)^2 - 1) / (m^2 (H, N)^2)}}{4m^2 (H, N)^4},$$

где  $H = (L+V) / \|L+V\|$  – нормализованный средний вектор между  $L$  и  $V$ ,  $m$  – среднеквадратичный наклон микрограней. Значение  $m$  задается в интервале от 0.2 для гладких материалов до 0.6 для шероховатых.

### 1.2. Моделирование самозатенения микрограней

Совокупность различно ориентированных граней образует микрорельеф, способный блокировать распространение некоторых свето-

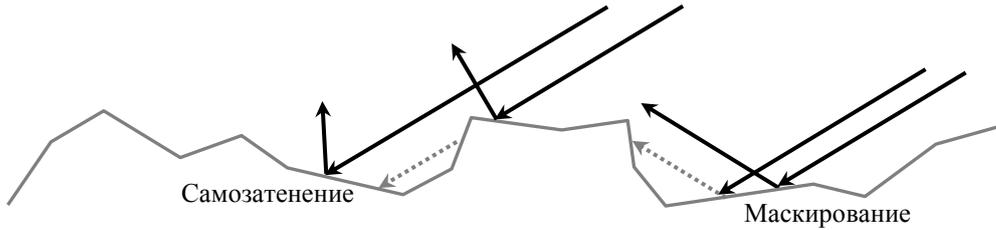


Рис. 3. Блокирование световых лучей микрорельефом поверхности

вых лучей, что приводит к эффектам самозатенения и маскирования (рис. 3). Первый возникает в том случае, когда выпуклости рельефа не дают падающим лучам осветить заслоненные этими выпуклостями участки, создавая в них тень. Маскирование происходит при попадании лучей, отраженных от одних микрограней, в другие грани. Таким образом, такие лучи, не распространяются в сторону наблюдателя. Для моделирования данных эффектов в формуле (1) применяется функция

$$G = \min \left( 1, \frac{2(\mathbf{H}, \mathbf{N})(\mathbf{V}, \mathbf{N})}{(\mathbf{V}, \mathbf{H})}, \frac{2(\mathbf{H}, \mathbf{N})(\mathbf{L}, \mathbf{N})}{(\mathbf{V}, \mathbf{H})} \right),$$

где  $\mathbf{N}$  – средняя нормаль к поверхности в рассматриваемой точке  $P$ ,  $\mathbf{L}$  – вектор из  $P$  на источник света,  $\mathbf{V}$  – вектор из  $P$  на наблюдателя,  $\mathbf{H} = (\mathbf{L} + \mathbf{V}) / \|\mathbf{L} + \mathbf{V}\|$  – нормализованный средний вектор между  $\mathbf{L}$  и  $\mathbf{V}$ .

### 1.3. Вычисление коэффициента Френеля

Кроме уже описанных выше параметров, рассматриваемая модель освещения учитывает также коэффициент Френеля  $F$ , который определяет долю света, отражаемого каждой микрогранью. Значение  $F$  зависит от угла падения светового луча и свойств преломления материала света с различными длинами волн. Для быстрого практического вычисления коэффициента Френеля в системах визуализации целесообразно использовать аппроксимацию Шлика, согласно которой доля отраженной в точке поверхности энергии рассчитывается по формулам

$$F = F_0 + (1 - F_0)(1 - (\mathbf{V}, \mathbf{H}))^5,$$

$$F_0 = \left( \frac{n_1 - n_2}{n_1 + n_2} \right)^2,$$

где  $F_0$  – коэффициент отражения при нормальном падении светового луча,  $n_1$  и  $n_2$  – показатели преломления материала объекта и внешней среды. Значения  $F_0$  для типовых материалов можно также найти в общедоступных таблицах.

## 2. Расширенная модель

Описанная модель Кука-Торренса позволяет с высокой степенью реалистичности визуализировать виртуальные объекты, сделанные из различных материалов. Однако она не включает в себя возможность моделирования теней от объектов, а также параметров источника освещения (кроме интенсивности). Кроме того, функция моделирования самозатенения микрограней в оригинальной модели никак не учитывает шероховатость поверхности, задаваемую значением параметра  $m$  среднеквадратичного наклона микрограней.

В данном исследовании предлагается расширенная модель расчета интенсивности света, отраженного в точке поверхности объекта виртуальной среды. Данная вычислительная модель решает указанные выше проблемы и эффективно реализуется на современных графических процессорах, обеспечивая распределенный рендеринг трехмерных сцен в масштабе реального времени. Определение интенсивности  $I_r$  отраженного освещения в некоторой точке  $P$  поверхности, согласно предлагаемой модели, выполняется по следующим формулам:

$$I_r = R_a I_a + \sum_{j=0}^n (F_j I_j \max(0, (\mathbf{L}_j, \mathbf{N}))) \times$$

$$\times \left( (1 - k_{sh,j})(1 - k_{met}) R_d + k_{sh,j} c_{sh} + f_{sh,j} k_{met} R_s \right),$$

$$k_{sh,j} = d_{sh,j} \cdot f_{sh,j},$$

где  $R_a$  – коэффициент отражения материалом рассеянного света;  $I_a$  – интенсивность рассеянного света в виртуальной сцене;  $n$  – общее число активных источников освещения;  $I_j$  – интенсивность света от  $j$ -го источника освещения;  $\mathbf{L}_j$  – единичный вектор из точки  $P$  на  $j$ -й источник света;  $\mathbf{N}$  – единичный вектор нормали в точке  $P$ ;  $k_{met}$  – доля зеркальной составляющей отражения;  $R_d$  и  $R_s$  – отражаемые материалом в точке  $P$  диффузная и зеркальная составляющие света (первая соответствует цвету материала, который обычно хранится в диффузной текстуре или текстуре альbedo);  $c_{sh}$  – цвет тени,  $d_{sh,j}$  – коэффициент плотности тени

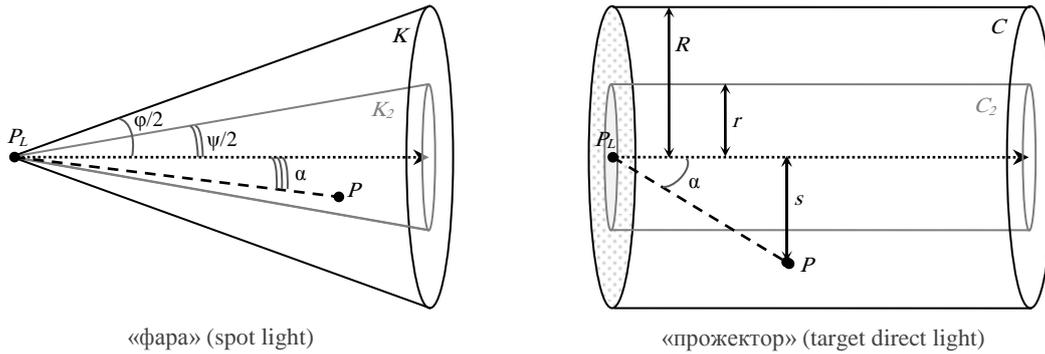


Рис. 4. Модели виртуальных светильников

(для  $j$ -го источника),  $f_{sh,j}$  – параметр, определяющий наличие или отсутствие тени в точке  $P$  (для  $j$ -го источника);  $F_\lambda$  – функция, учитывающая различные типы и параметры источника света.

Значение  $f_{sh}$  для каждого источника света в каждой рассчитываемой точке (пикселе) определяется во фрагментном шейдере с использованием технологии карт теней (*shadow maps*), подробно изложенной в работе [6]. Далее рассмотрим вычисление  $R_s$  и  $F_\lambda$ .

### 2.1. Вычисление зеркального отражения

Зеркальная составляющая  $R_s$  отражения света от поверхности объекта в некоторой ее точке  $P$  в предлагаемом решении вычисляется аналогично модели Кука-Торренса, то есть по формуле (1). Однако используемая в исходной модели функция  $G$  распределения наклона микрограней (см. п. 1.2) не удовлетворяет экспериментальным данным, а ее значение не зависит от того, гладкая поверхность или шероховатая. Для решения данной проблемы в качестве  $G$  будем использовать аппроксимацию (с высокой точностью) уравнения Смита, которое лишено указанных недостатков. Тогда

$$G = \frac{(V, N)}{(1-k)(V, N) + k} \cdot \frac{(L, N)}{(1-k)(L, N) + k},$$

$$k = \sqrt{\frac{2m^2}{\pi}},$$

где  $m$  – среднеквадратичный наклон микрограней (см. п. 1.1).

### 2.2. Моделирование источников света

В разработанной расширенной модели освещения используются три типа источников света, распространенные в системах компьютерного моделирования: *omni light* (всенаправленный), *spot light* («фара») и *target direct light* («прожектор»). Отметим, что модель позволяет легко добавлять и другие типы освеще-

тителей. Источники *omni* излучают свет одинаково во всех направлениях, поэтому для них

$$F_\lambda = F_{att}(d),$$

где  $F_{att}$  – функция, вычисляющая коэффициент затухания интенсивности света в зависимости от расстояния  $d$  между источником и рассматриваемой точкой  $P$  объекта. В предлагаемой модели затухание света реализуется с использованием обратно-квадратичной функции

$$F_{att}(d) = \frac{1}{k_{quad}d^2 + k_{lin}d + k_{const}},$$

где  $k_{quad}$ ,  $k_{lin}$  и  $k_{const}$  – квадратичный, линейный и постоянный коэффициенты затухания.

Освещение от источников типа «фара» распространяется в пределах некоторого конуса  $K$  с углом раствора  $\phi < \pi$  (рис. 4, слева). Данный источник также характеризуется углом  $\psi$  «горячего пятна», который задает коническую область  $K_2$ , вложенную в  $K$ . Внутри  $K_2$  интенсивность освещения от источника имеет максимальное значение  $I$ , равное осевой силе света, а между  $K_2$  и  $K$  убывает от  $I$  до 0. При этом значение функции  $F_\lambda$  для некоторой точки  $P$  поверхности объекта зависит от угла  $\alpha$  между направлением свечения и отрезком, соединяющим  $P$  с точкой  $P_L$  расположения источника. Учитывая затухание света, определим функцию  $F_\lambda$  для источника «фара», как

$$F_\lambda = \begin{cases} F_{att}(d), & \text{при } \alpha \leq \psi/2 \\ \frac{\cos \alpha - \cos(\phi/2)}{\cos(\psi/2) - \cos(\phi/2)} F_{att}(d), & \text{при } \alpha \in (\psi/2, \phi/2] \\ 0, & \text{при } \alpha > \phi/2 \end{cases}.$$

Источники типа «прожектор» светят в пределах полубесконечного цилиндра  $C$ , имеющего радиус  $R$  (рис. 4, справа). Аналогично случаю с «фарами», в световом потоке данного типа источников присутствует область с максимальной интенсивностью  $I$  освещения, пред-

ставляющая собой вложенный в  $C$  цилиндр  $C_2$  с радиусом  $r$ . Оси обоих цилиндров совпадают. Между  $C_2$  и  $C$  интенсивность света падает от  $I$  до 0.

Значение функции  $F_\lambda$  для «прожектора» имеет зависимость от расстояния  $s$  между точкой  $P$  объекта и осью цилиндра.

С учетом затухания света получаем следующую формулу:

$$F_\lambda = \begin{cases} F_{att}(d), & \text{при } s \leq r, \alpha \leq \pi/2 \\ \frac{R-s}{R-r} F_{att}(d), & \text{при } s \in (r, R], \alpha \leq \pi/2, \\ 0, & \text{при } s > R \text{ или } \alpha > \pi/2 \end{cases}$$

где  $\alpha$  – угол между направлением свечения и отрезком, соединяющим  $P$  с точкой  $P_L$  расположения источника. Значение  $\alpha > \pi/2$  соответствует нахождению рассматриваемой точки  $P$  позади источника.

## Заключение

В данном исследовании предложена оригинальная модель расчета интенсивности света, отраженного в точке поверхности объекта трехмерной виртуальной среды. Данная модель обеспечивает имитацию физически корректного освещения при выполнении рендеринга виртуальных сцен, а также поддерживает распределенную реализацию такого рендеринга с использованием современных многоядерных графических процессоров в масштабе реального времени.

На основе предложенных в работе решений были созданы программные модули, которые прошли успешную апробацию в составе системы визуализации трехмерных виртуальных сцен, разработанной в ФГУ ФНЦ НИИСИ РАН. Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-07-00950.

# Reflected light calculation model for imitation of physically correct lighting of virtual environment objects on GPU

A.V. Maltsev, D.V. Omelchenko

**Abstract:** The paper proposes original computational model for distributed intensity calculation of light reflected from object surface of three-dimensional virtual environment. This solution is an extension of the Cook-Torrance model and allows to realize different types of light sources and shadows casted by objects in virtual scenes. Created model provides a capability of distributed real-time rendering of virtual environment with imitation of physically correct lighting on multicore graphics processors.

**Keywords:** visualization, material, distributed computing, lighting model, graphics processor.

## Литература

1. B.T. Phong. Illumination for computer generated pictures // Communications of ACM 18, 1975, № 6, pp. 311-317.
2. J.F. Blinn. Models of light reflection for computer synthesized pictures // Proceedings of 4th annual conference on computer graphics and interactive techniques, 1977, pp. 192-198.
3. T. Ritschel, C. Dachsbacher, T. Grosch, J. Kautz. The state of the art in interactive global illumination // Computer Graphics Forum, 2012, vol. 31(1), pp. 160-188.
4. R.L. Cook, K.E. Torrance. A reflectance model for computer graphics // ACM Transactions on Graphics, 1982, № 1(1), pp. 7-24.
5. Katsushi Ikeuchi. Lambertian Reflectance // Encyclopedia of Computer Vision, Springer, 2014, pp. 441-443.
6. Мальцев А.В. Применение теневых карт для моделирования теней в виртуальных 3D сценах в реальном времени // Труды НИИСИ РАН, 2014, т. 4, № 2, с. 69-77.

# Технология моделирования и управления источниками искусственного освещения МКС в системах виртуального окружения

А.В. Мальцев<sup>1</sup>, П.Ю. Тимохин<sup>2</sup>, Д.В. Омельченко<sup>3</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

E-mail's: <sup>1</sup>avmaltcev@mail.ru, <sup>2</sup>webpismo@yahoo.de, <sup>3</sup>omelchenko\_dv@mail.ru

**Аннотация:** В работе предлагается новая технология и алгоритм управления виртуальными моделями источников подсветки внешней поверхности Международной космической станции, входящих в состав специализированных блоков видеонаблюдения ETVCG. Также рассматриваются методы моделирования данных блоков в системах виртуального окружения. Реализация источников, входящих в состав ETVCG, выполняется на основе комбинации осветителей типа «фара» (spotlight).

**Ключевые слова:** управление, моделирование, виртуальное окружение, МКС, источник света.

## Введение

В настоящее время во многих сферах человеческой деятельности широко используется компьютерное моделирование трехмерных сцен, при котором реальная окружающая среда и ее объекты заменяются визуализируемыми виртуальными аналогами. Такой подход применяется, в том числе, в системах виртуального окружения, связанных с задачами подготовки специалистов для работы на Международной космической станции [1-3]. Это позволяет на Земле отрабатывать навыки их поведения в штатных и нештатных ситуациях, возникающих в ходе космических экспедиций на МКС. Эффективность такой подготовки во многом определяется тем, насколько синтезируемая виртуальная среда схожа со своим реальным прототипом. Ключевыми факторами здесь являются степень детализации объектов виртуального окружения и наличие у оператора возможностей отработки навыков управления ими.

В данной работе предлагается новая технология и алгоритм управления источниками подсветки внешней поверхности МКС в системах виртуального окружения. Для этого была создана трехмерная виртуальная модель специализированного блока видеонаблюдения ETVCG (*External Television Camera Groups*, [4]), который включает в себя указанный источник, а также был разработан оригинальный метод моделирования освещения объек-

тов таким источником. Далее рассмотрим эти решения подробнее.

## 1. Источники подсветки МКС

Выполнение работ, связанных с выходом космонавтов в открытый космос, или проведение каких-либо дистанционно управляемых роботизированных операций, сопровождается визуальным контролем происходящего с помощью видеокамер. При этом необходимо иметь хорошую освещенность рабочей области на поверхности МКС. В условиях недостаточного естественного освещения (темное время суток, самозатенение элементами конструкции станции и т.п.) требуемый уровень освещенности обеспечивается источниками света VCL (*video camera luminaire*, рис. 1), входящими в состав устройств видеонаблюдения ETVCG. Блоки ETVCG (рис. 2, 3), включающие в себя также видеокамеру, размещаются на длинных штангах, которые закреплены на поверхности модулей МКС в специально отведенных для этого местах – портах камер (*camera port*, CP).

VCL имеет овальный корпус, внутри которого расположено две лампы с коническими рефлекторами (рис. 3). Включение ламп чередуется каждый раз при подаче напряжения на светильник, т.е. загорается та лампа, которая в предыдущий раз не горела. Такой подход обеспечивает наличие резервного источника света при выходе из

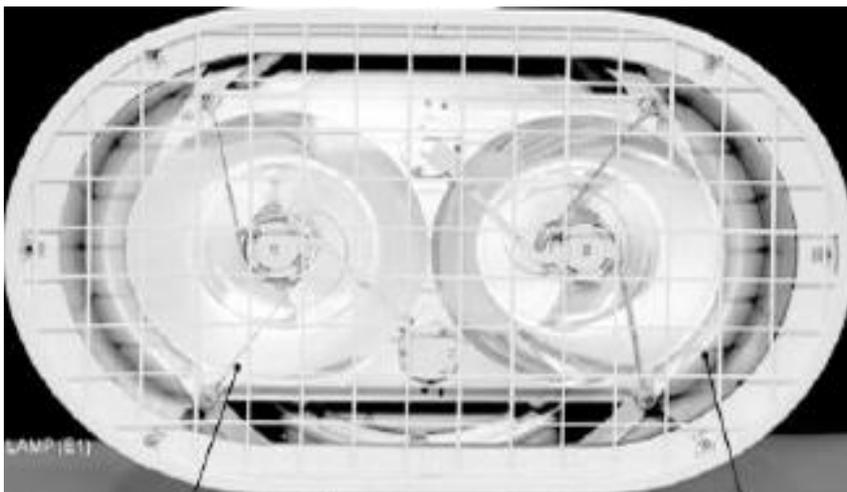


Рис. 1. Источник света VCL



Рис. 2. Подсветка внешней поверхности МКС с помощью осветителя в блоке ETVCG

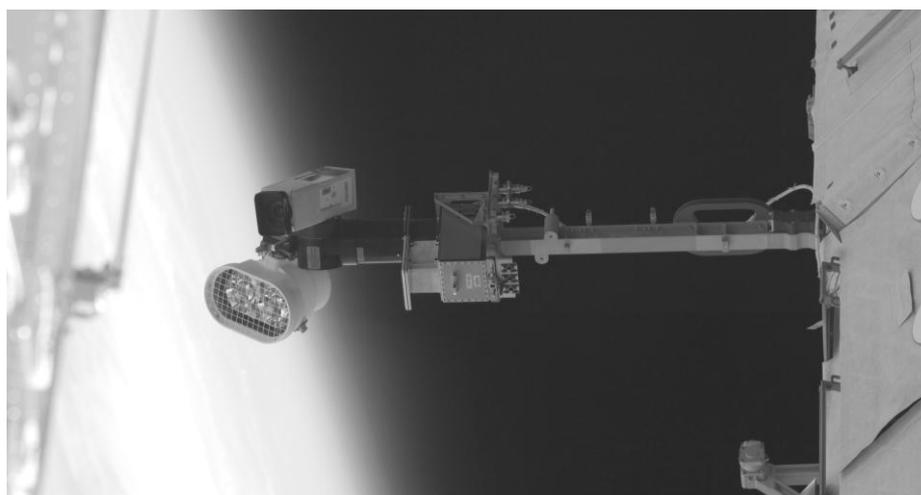


Рис. 3. Управляемый блок ETVCG с источником света

строения одной из ламп во время выполнения визуального контроля проводимых на поверхности МКС работ, а также продлевает срок службы каждой лампы.

Каждый блок ETVCG является управляемым. Для выбора направления подсветки и

взгляда камеры производится регулировка углов панорамирования и наклона. Возможность управления данными углами обеспечивается системой из пары одноосных шарниров. При этом происходит синхронное изменение ориентации источника VCL и видеокамеры

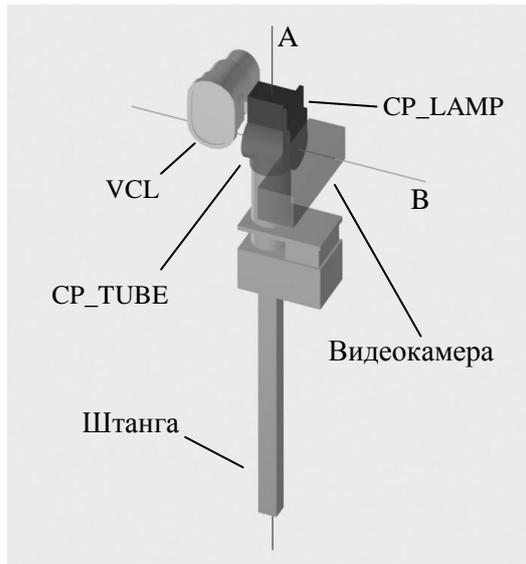


Рис. 4. Компоненты модели ETVCG

путем поворотов этих устройств вокруг продольной оси штанги (панорамирование), а также в параллельных этой оси плоскостях (наклон). Управление яркостью VCL выполняется путем настройки значений осевой силы света и силы света на границе так называемого полезного угла.

## 2. Модель источника света в составе блока ETVCG

С целью разработки технологии управления моделями описанных источников в системах виртуального окружения, в данном исследовании была создана собственная подвижная трехмерная виртуальная модель блока ETVCG и предложен метод реалистичной имитации освещения от входящего в его состав источника света VCL. В данной модели реализованы описанные в разделе 1 возможности управления поворотами VCL и камеры. Они осуществляются с помощью вращения

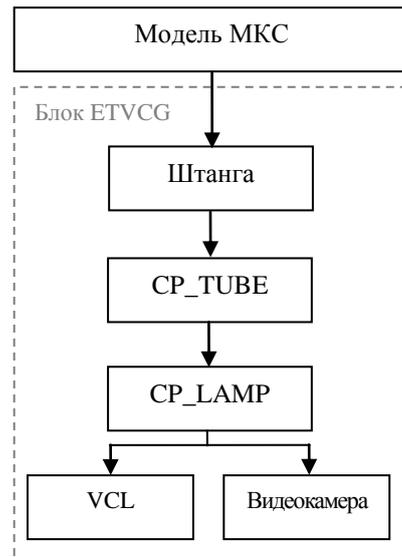


Рис. 5. Иерархическая структура модели

объектов CP\_LAMP и CP\_TUBE (схематично изображены на рис. 4), расположенных между штангой и корпусом источника. Оба этих объекта, а также камера и VCL входят в жестко связанную иерархическую структуру, представленную на рисунке 5. Управление положением или ориентацией какого-либо узла в такой структуре влечет за собой соответствующие изменения положений и ориентаций дочерних узлов. Таким образом, поворот CP\_TUBE на угол  $\alpha$  вокруг продольной оси A штанги осуществляет панорамирование корпуса источника и видеокамеры. Наклон VCL и камеры обеспечивается поворотом объекта CP\_LAMP на угол  $\beta$  вокруг перпендикулярной штанге оси B, связанной с CP\_TUBE (см. рис. 4).

Осветитель VCL создает конический световой поток с углом раствора конуса около  $50^\circ$ , характеризующийся определенными значениями осевой силы света  $I_a$  и силы света  $I_e$  на границе так называемого полезного угла  $\gamma$  (рис. 6а). За пределами конуса освещение

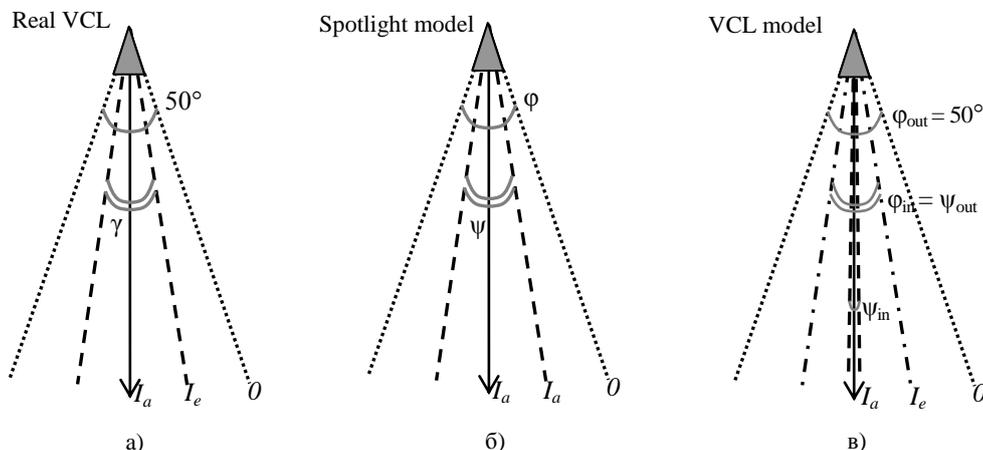


Рис. 6. Моделирование VCL на основе двух источников типа «фара»



Рис. 7. Визуализация виртуальной модели МКС с работающим источником в порту камер CP13

отсутствует, т.е. сила света равна 0. Моделирование освещения от VCL в данном исследовании осуществляется на основе комбинации двух источников освещения одного распространенного типа – направленного виртуального источника spotlight («фара»). Область действия для такого источника управляется с помощью двух параметров:  $\varphi$  и  $\psi$ . Значение  $\varphi$  определяет угол раствора конуса, в пределах которого распространяется освещение, а значение  $\psi$  – так называемый угол «горячего пятна», т.е. конус, внутри которого освещение от источника «фара» имеет постоянную силу света, равную осевой  $I_a$  (рис. 6б). При этом  $0 < \psi \leq \varphi < 180^\circ$ . Вне конуса с углом  $\varphi$  сила света равна 0, а между конусами она равномерно изменяется от 0 до  $I_a$ .

В предлагаемой комбинированной модели освещения обе «фары» размещаются в одной и той же точке – центре лампы виртуального источника VCL, и для них задаются одинаковые векторы направления освещения, совпадающие с направлением оси света VCL (рис. 6в). Конусы, определяемые углами  $\varphi$  и  $\psi$  каждой из «фар» (внутренней *in* и внешней *out*), имеют вложенную структуру (см. рис. 6в), описываемую следующими параметрами:

- $\varphi_{out} = 50^\circ$  – угол раствора конуса первой (внешней) «фары», задающий границу светового потока от осветителя VCL;
- $\varphi_{in} = \psi_{out} = \gamma$  – угол раствора конуса второй (внутренней) «фары» (он же угол «горячего пятна» внешней «фары»), задающий полезный угол излучения осветителя VCL;
- $\psi_{in} = 2^\circ$  – угол «горячего пятна» внутренней «фары», имитирующий небольшое пятно пикового освещения реального источника VCL.

В описанной комбинированной модели сила  $I_{a,out}$  света «горячего пятна» внешней «фары» задается равной силе  $I_e$  света на границе полезного угла осветителя VCL, а сила  $I_{a,in}$  света «горячего пятна» внутренней «фары» рассчитывается как

$$I_{a,in} = I_a - I_{a,out},$$

где  $I_a$  – осевая сила света реального VCL.

За пределами своего конуса (включая его поверхность) внутренняя «фара» не излучает свет (ее сила света равна 0), поэтому сила света комбинации двух «фар» в этой области будет уменьшаться от  $I_{a,out}$  до 0, что соответствует изменению сила света VCL от  $I_e$  до 0. Внутри же этого конуса, благодаря принципу аддитивности, результирующая сила света комбинации двух «фар» будет уменьшаться от значения  $(I_{a,in} + I_{a,out})$  до  $I_{a,out}$ , что соответствует изменению сила света VCL от  $I_a$  до  $I_e$ .

На рисунке 7 представлен пример кадра визуализации виртуальной модели Международной космической станции с установленным в порт камер CP13 блоком ETVCG, подсвечивающим поверхность модуля станции. Расчет освещенности от источника VCL, входящего в данный блок, был выполнен на основе описанного выше подхода.

### 3. Управление источниками VCL

Чтобы реализовать в системах виртуального окружения управление моделью блока ETVCG и входящего в него осветителя VCL, в данной работе предлагается новая технология, в которой управляющие команды генерируются вычислительным комплексом (в соответствии с сигналами от пульта управления, за которым сидит оператор) и с интервалами 40 мс посылаются в систему визуализации (где уже загружена виртуальная модель МКС) посредством сети TCP/IP с использованием специально разработанного протокола передачи данных (рис. 8). Этот протокол включает в себя ряд типов управляющих команд, а именно:

- MOV – управление ориентацией виртуального объекта;
- LGA – управление параметрами  $\varphi$  и  $\psi$  источника света типа spotlight;



Рис. 8. Структура системы виртуального окружения МКС

- FLP – установка значения вещественного управляемого параметра;
- VAL – установка значения целочисленного управляемого параметра.

Управление моделью блока ETVCG и светильником VCL осуществляется по следующему алгоритму:

1. Оператор с помощью реального или виртуального пульта управления изменяет значение необходимого параметра (угла панорамирования  $\alpha$ , угла наклона  $\beta$ , полезного угла излучения  $\gamma$ , значения силы света  $I_{a,ин}$  или  $I_e$ ), которое передается в вычислительный комплекс;
2. Вычислительный комплекс формирует сетевой пакет с управляющей командой, включающий тип команды, символьный идентификатор управляемого параметра объекта, числовое значение параметра, и передает его в систему визуализации;
3. Система визуализации, получив пакет с управляющей командой, выполняет его разбор и изменяет указанный параметр в виртуальной модели в соответствии с записанным в пакете значением.

В данной работе проводилось моделирование трех блоков ETVCG, установленных на МКС в порты камер CP3, CP8 и CP13. В таблице 1 представлены управляемые пара-

метры блока ETVCG с источником света в порту CP3, соответствующие этим параметрам типы команд, идентификаторы и числовые значения. Для блоков в портах CP8 и CP13 эти данные аналогичны, отличие заключается лишь в цифрах идентификаторов.

## Заключение

В данном исследовании предложены технологии моделирования и управления виртуальными источниками VCL искусственного освещения, которые входят в состав блоков видеонаблюдения ETVCG, размещенных на поверхности МКС. Адекватность описанных решений была подтверждена в ходе их апробации в рамках комплекса виртуального окружения, разработанного в ФГУ ФНЦ НИИСИ РАН. Полученные результаты также в других графических приложениях, требующих детального и реалистичного рендеринга модели МКС.

Публикация выполнена в рамках государственного задания по проведению фундаментальных научных исследований (ГП 14) по теме (проекту) «34.9. Системы виртуального окружения: технологии, методы и алгоритмы математического моделирования и визуализации». (0065-2018-0020).

Таблица 1. Управление источником света в составе блока ETVCG

Управляемый параметр источника света в блоке ETVCG (CP3)	Тип команды	Идентификатор	Значение
Включение/выключение источника	VAL	CP3	0/1
Разность между осевой силой света и силой света на границе полезного угла, $I_{a,ин}$	FLP	LDIFF_CP3	Вещественное, кд
Сила света на границе полезного угла, $I_e$	FLP	LEDGE_CP3	Вещественное, кд
Угол панорамирования, $\alpha$	MOV	CP3_TUBE	0.0°...360.0°
Угол наклона, $\beta$	MOV	CP3_LAMP	-90.0°...90.0°
Полезный угол излучения, $\gamma$	LGA	CP3	5.0°...49.0°

# The technology of modeling and controlling light sources of the ISS in virtual environment systems

A.V. Maltsev, P.Yu. Timokhin, D.V. Omelchenko

**Abstract:** The paper proposes new technology and algorithm for controlling virtual models of light sources on the outer surface of the International Space Station, which are part of specialized ETVCG video surveillance units. Also the methods for such unit simulation in virtual environment systems are considered. The implementation of light sources included to ETVCG is performed by means of the combination of two spotlights.

**Keywords:** controlling, simulation, virtual environment, the ISS, light source.

## Литература

1. М.В. Михайлюк, В.И.Брагин. Технологии виртуальной реальности в имитационно-тренажерных комплексах подготовки космонавтов // Пилотируемые полеты в космос. – 2013. – № 2. – С. 82-93.
2. М.В.Михайлюк, И.Н.Мироненко. Моделирование антропоморфных роботов в космических тренажерах // Abstracts of Twelfth International Seminar on Mathematical Models & Modeling in Laser-plasma Processes & Advanced Science Technologies. – 2014, p. 69.
3. А.И.Масалкин, М.А.Торгашев. Опыт использования систем имитации визуальной обстановки в тренажерах пилотируемых космических аппаратов // Пилотируемые полеты в космос. – 2015. – № 2. – С. 36-42.
4. Does the International Space Station Have External Lights? // URL: [https://www.huffingtonpost.com/entry/does-the-international-space-station-have-external\\_us\\_58ed3198e4b0ea028d568d9a](https://www.huffingtonpost.com/entry/does-the-international-space-station-have-external_us_58ed3198e4b0ea028d568d9a) (дата обращения: 24.09.2018).

# Моделирование динамики манипуляционных роботов в системах виртуального окружения

Е.В. Страшнов<sup>1</sup>, Л.А. Финагин<sup>2</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия, E-mail's: <sup>1</sup>strashnov\_evg@mail.ru, <sup>2</sup>finagin\_la@mail.ru

**Аннотация:** Рассматривается задача моделирования в реальном времени динамики манипуляционных роботов в системах виртуального окружения. Предлагаемое решение этой задачи состоит в применении метода составных тел и его модификаций на основе импульсов, что позволяет реализовать динамику сложных кинематических механизмов и захватных устройств манипуляторов. При этом обеспечение выполнения ограничений осуществляется путем решения задачи MLCP (задачи со смешанными линейными дополнениями) относительно неизвестных значений импульсов.

**Ключевые слова:** система шарнирно связанных тел, реальное время, метод составных тел, импульсный подход, задача MLCP, системы виртуального окружения

## Введение

В настоящее время актуальным направлением научных исследований является имитационное моделирование манипуляционных роботов и механизмов в системах виртуального окружения. Манипуляционный робот [1] представляет собой техническое устройство (подвижное или стационарное), которое состоит из механической части (один или несколько манипуляторов) для выполнения двигательных функций, аналогичным функциям рук человека, и оснащен рабочим органом. Манипулятор робота служит для выполнения различных технологических операций, в которых требуется захват, перемещение и ориентирование объектов.

С применением систем виртуального окружения осуществляется решение некоторых задач, связанных с управлением манипуляционных роботов. Примером служит дистанционное управление моделями роботов в виртуальной среде с помощью пультов управления (реальных или виртуальных). Такой подход позволяет обучить человека-оператора навыкам управления реальными роботами и в дальнейшем уменьшить риск их поломки. В свою очередь реализация автоматического управления роботами в виртуальном окружении на основе функциональных схем облегчает тестирование и сокращает время и усилия при создании реальных систем управления. Кроме того, существуют задачи в космической отрасли, которые невозможно или трудно воспроизвести в земных условиях. К таким задачам относится исследование поверхности других планет и выполнение на них различных

работ с помощью манипуляционных роботов. В этом случае отработка технологических операций в виртуальной среде может дать правильную экспертную оценку о применимости роботов для решения этих задач. При этом важно, чтобы у пользователей систем виртуального окружения не возникло ложных представлений о движении и поведении манипуляционных роботов. Для этого необходимо, чтобы моделирование динамики роботов осуществлялось в режиме реального времени с применением адекватных физических моделей. Поэтому задача, решаемая в данной работе, состоит в разработке методов и алгоритмов моделирования динамики манипуляционных роботов, которые будут обеспечивать необходимую точность и качество.

Для моделирования динамики манипуляционных роботов предлагается задействовать метод составных тел [2, 3]. Этот метод базируется на использовании минимального набора координат (обобщенных координат) и предназначен для решения задачи прямой динамики моделирования шарнирно связанных тел со структурой в виде дерева. В работе [4] был разработан алгоритм на основе метода последовательных импульсов, который позволяет обеспечивать выполнение ограничений различного вида. Здесь предлагается подход, в котором ограничения обрабатываются одновременно с формулировкой и решением задачи MLCP [5] (Mixed Linear Complementarity Problem – задача со смешанными линейными дополнениями). Предлагаемое решение позволяет в режиме реального времени моделировать сложные кинематические структуры манипуляторов, трение и ограничения на параметры в шарнирах, столкновения звеньев манипулято-

ра и других объектов с реализацией удара, контакта и различных видов трения.

## 1. Динамика систем шарнирно связанных тел

Рассмотрим сначала систему шарнирно связанных тел со структурой в виде дерева, где шарниры имеют одну степень свободы (осевые или призматические). Тогда эта система тел описывается с помощью упорядоченного набора обобщенных координат  $\mathbf{q}$  размерности  $N$ , где  $N$  – количество шарниров.

Задача прямой динамики для системы связанных тел состоит в вычислении обобщенных ускорений  $\ddot{\mathbf{q}}$  через внешние силы и моменты  $\mathbf{F}^{(e)}$  и обобщенные силы в шарнирах  $\boldsymbol{\tau}$ . Это описывается с помощью функции

$$\ddot{\mathbf{q}} = FD(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{F}^{(e)}, \boldsymbol{\tau}). \quad (1)$$

Задача (1) на каждом шаге  $\Delta t$  моделирования динамики решается с помощью рекурсивного метода составных тел [2]. Для интегрирования уравнений движения применим полунявную схему Эйлера, в которой сначала определяются обобщенные скорости, а затем обобщенные координаты:

$$\dot{\mathbf{q}}(t + \Delta t) = \dot{\mathbf{q}}(t) + \ddot{\mathbf{q}}(t)\Delta t; \quad (2)$$

$$\mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \dot{\mathbf{q}}(t + \Delta t)\Delta t. \quad (3)$$

Таким образом, моделирование динамики систем шарнирно связанных тел со структурой в виде дерева состоит в решении задач (1) – (3) для каждого момента времени  $t + \Delta t$ .

Однако в общем случае моделирование динамики систем тел предполагает обработку произвольных кинематических структур тел, коллизий, трения и т.д. Это приводит к необходимости в формулировке и выполнении ограничений, которые могут формироваться как через абсолютные, так и через относительные координаты.

Рассмотрим формулировку задачи моделирования динамики систем связанных тел на примере ограничений, накладываемых на обобщенные (относительные) координаты. Такие ограничения описываются с помощью функции вида

$$G = G(q_m, q_l), \quad m, l \in [1, N], \quad (4)$$

где  $G = 0$  соответствует ограничению в виде равенства,  $G \geq 0$  – в виде неравенства.

Ограничения на обобщенные скорости системы шарнирно связанных тел характеризуются функцией

$$\chi = \chi(\dot{q}_m, \dot{q}_l), \quad m, l \in [1, N], \quad (5)$$

где  $\chi = 0$  соответствует ограничению в виде равенства,  $\chi \geq 0$  – в виде неравенства. Для голономных связей функция (5) может быть получена путем дифференцирования (4) по времени.

В данной работе предлагается импульсный подход для обеспечения выполнения ограничений вида (4) и (5). Известно [5], что моделирование динамики систем тел с ограничениями в общем случае формулируется в виде задачи MLCP (задача со смешанными линейными дополнениями) относительно неизвестных  $\mathbf{w}$  и  $\mathbf{p}$ :

$$\begin{aligned} \mathbf{w} &= \mathbf{A}\mathbf{p} + \mathbf{b}, \quad \mathbf{p}_{\min} \leq \mathbf{p} \leq \mathbf{p}_{\max}, \quad \mathbf{p}, \mathbf{w}, \mathbf{b} \in \mathbb{R}^n; \\ p_i &= p_{\min,i} : w_i \geq 0; \\ p_i &= p_{\max,i} : w_i \leq 0; \\ p_{\min,i} < p_i < p_{\max,i} : w_i &= 0, \quad i = \overline{1, n}, \end{aligned} \quad (6)$$

где  $\mathbf{A}$  – симметричная положительно определенная матрица,  $\mathbf{p}$  – вектор компонентов импульсов,  $\mathbf{w}$  – вектор скоростей ограничений после применения импульсов,  $n$  – число ограничений. Границы  $\mathbf{p}_{\min}$  и  $\mathbf{p}_{\max}$  ограничений на компоненты импульсов в общем случае могут зависеть от импульсов  $\mathbf{p}$ .

Динамика систем шарнирно связанных тел с ограничениями приводит к составлению и решению задачи (6). Опишем сначала способ вычисления матрицы  $\mathbf{A}$  и вектора  $\mathbf{b}$  первого уравнения в (6).

Вектор  $\mathbf{b}$  задачи (6) определяется через текущие скорости после применения известных сил и моментов в (1) и (2). Обозначим обобщенные скорости после применения сил как  $\dot{\mathbf{q}}(t + \Delta t)$ . Тогда через эти скорости соотношение (2) переписывается в следующем виде

$$\dot{\mathbf{q}}(t + \Delta t) = \dot{\mathbf{q}}(t) + \ddot{\mathbf{q}}(t)\Delta t, \quad (7)$$

а для ограничений вида (5) будет выполнено  $b_i = \chi_i \dot{q}_m(t + \Delta t), \dot{q}_l(t + \Delta t)$ .

Компоненты матрицы  $\mathbf{A}$  задачи (6) вычисляются на основе технологии применения единичного тестового импульса  $p'_i = 1$  для определения изменений скоростей. Для этого в [4] был разработан алгоритм, являющийся модификацией метода составных тел, в котором изменение обобщенных скоростей  $\Delta \dot{\mathbf{q}}$  определяются через импульсы сил  $\mathbf{p}_f$  и обобщенные импульсы в шарнирах  $\mathbf{p}_\tau$ . Это описывается с помощью следующей функции:

$$\Delta \dot{\mathbf{q}} = FC(\mathbf{q}, \mathbf{p}_f, \mathbf{p}_\tau). \quad (8)$$

Если ограничение накладывается на абсолютные координаты тел, то формируется тестовый импульс с компонентой  $p_{f,i}^t = 1$ , в противном случае  $p_{f,i}^t = 1$ . Идея предлагаемого подхода состоит в том, что, применяя тестовый импульс  $p_i = 1$  с  $p_j = 0, j = \overline{1, n}, j \neq i$ , компоненты  $i$ -го столбца матрицы  $\mathbf{A}$  вычисляются следующим образом  $a_{ji} p_i^t = a_{ji} = \Delta w_j$ , где  $\Delta w_j$  – скорость  $j$ -го ограничения после применения тестового импульса в (8). Для ограничения (5) после применения импульса  $p_{f,i}^t = 1$  получим, что компоненты симметричной матрицы  $\mathbf{A}$  вычисляются как

$$a_{ij} = a_{ji} = \chi_j(\Delta \dot{q}_m, \Delta \dot{q}_i), \quad i, j = \overline{1, n}.$$

Задача (6) решается с помощью прямых или итерационных методов. В дальнейшем мы опишем основные положения проекционного метода Гаусса-Зейделя для итерационного решения этой задачи.

После вычисления компонентов импульсов  $\mathbf{p}$  из (6) и применения соотношения (8) обобщенные скорости корректируются как

$$\dot{\mathbf{q}}(t + \Delta t) = \check{\dot{\mathbf{q}}}(t + \Delta t) + FC(\mathbf{q}(t), \mathbf{p}). \quad (9)$$

Таким образом, моделирование динамики систем шарнирно связанных тел с произвольной структурой и наличием ограничений для момента времени  $t + \Delta t$  сводится к вычислению обобщенных координат, их скоростей и ускорений с помощью (1), (7), (9) и (3) с решением задачи MLCP вида (6).

Для обеспечения выполнения ограничений вида (4) применяется стабилизация на основе метода отдельных импульсов. С основными положениями этого подхода можно ознакомиться в [4] и [6].

## 2. Проекционный метод Гаусса-Зейделя

Проекционный метод Гаусса-Зейделя представляет собой итерационный подход для решения задачи (6). Этот метод является модификацией традиционного метода Гаусса-Зейделя [7, 8], предназначенного для итерационного решения систем линейных алгебраических уравнений.

Пусть  $\mathbf{p}^{(s)}$  и  $\mathbf{w}^{(s)}$  – значения импульсов и скоростей ограничений на  $s$ -ой итерации. В проекционном методе Гаусса-Зейделя первое уравнение из (6) записывается в следующем виде

$$\sum_{j=1}^{j < i} a_{ij} p_j^{(s+1)} + a_{ii} p_i^{(s+1)} + \sum_{j=i+1}^n a_{ij} p_j^{(s)} + b_i = w_i^{(s+1)}, \quad i = \overline{1, n}. \quad (10)$$

Рассматриваемый подход для решения задачи (6) на  $s+1$ -ой итерации заключается в следующем. В начале решаем уравнение (10) относительно импульса  $p_i^{(s+1)}$  с  $w_i^{(s+1)} = 0$ . Обозначая этот импульс через  $\tilde{p}_i^{(s+1)}$ , получим для него следующее соотношение

$$\tilde{p}_i^{(s+1)} = (-\sum_{j=1}^{j < i} a_{ij} p_j^{(s+1)} - \sum_{j=i+1}^n a_{ij} p_j^{(s)} - b_i) / a_{ii},$$

где для положительно определенной матрицы  $\mathbf{A}$  ее диагональные элементы  $a_{ii} > 0$ .

Если  $p_{i, \min} \leq \tilde{p}_i^{(s+1)} \leq p_{i, \max}$ , то решением (6) будет  $p_i^{(s+1)} = \tilde{p}_i^{(s+1)}$  и  $w_i^{(s+1)} = 0$ . В противном случае, если  $\tilde{p}_i^{(s+1)} < p_{i, \min}$ , то  $p_i^{(s+1)} = p_{i, \min}$  и  $w_i^{(s+1)} = a_{ii}(p_{i, \min} - \tilde{p}_i^{(s+1)}) > 0$ . В том случае, если  $\tilde{p}_i^{(s+1)} > p_{i, \max}$ , то решением (6) будет  $p_i^{(s+1)} = p_{i, \max}$  и  $w_i^{(s+1)} = a_{ii}(p_{i, \max} - \tilde{p}_i^{(s+1)}) < 0$ .

Итерации проекционного метода Гаусса-Зейделя продолжают до тех пор, пока для каждого значения импульса  $p_i^{(s+1)}$  не будет выполнено условие

$$|p_i^{(s+1)} - p_i^{(s)}| < \varepsilon,$$

где  $\varepsilon$  – заданное число.

## 3. Моделирование динамики манипуляционных роботов

Моделирование динамики манипуляционных роботов с помощью метода составных тел включает в себя реализацию замкнутых кинематических механизмов, электродвигателей, шарниров с трением и ограничениями на их параметры, а также коллизии с ударом, контактом и трением звеньев манипулятора. Опишем эти составляющие более подробно с описанием ограничений, накладываемых на координаты (4), скорости (5) и импульсы в рамках задачи MLCP вида (6).

### Замкнутые кинематические цепи.

Некоторые механизмы захватного устройства манипулятора могут содержать систему шарнирно связанных тел со структурой в виде замкнутой кинематической цепи. В методе составных тел эта структура преобразуется к структуре в виде дерева с добавлением замыкающих шарниров. Тогда поведение этих шарниров заменяется ограничениями, накладываемыми на абсолютные координаты со-

единяемых шарниром звеньев. Мы рассматриваем такие механизмы захватного устройства, движение которых осуществляется в одной плоскости. В этом случае достаточно обеспечить выполнение ограничений на поступательные степени свободы, в то время как ограничения на вращательные степени свободы будут выполнены автоматически. Тогда получим следующие ограничения:

$$\mathbf{r}_p^m - \mathbf{r}_p^l = \mathbf{0}, \quad m, l \in [1, N],$$

где  $\mathbf{r}_p^m$  и  $\mathbf{r}_p^l$  – координаты положения замыкающего шарнира двух звеньев.

Путем дифференцирования данного соотношения получим ограничения относительно скоростей звеньев:

$$\mathbf{v}_p^m - \mathbf{v}_p^l = \mathbf{0}, \quad m, l \in [1, N].$$

#### Двигатели и повторители в шарнирах.

Электрические двигатели предназначены для управления относительным движением в шарнирах. В методе составных тел моделирование двигателей осуществляется путем расчета крутящих моментов [9]:

$$\tau_{en,i} = M_{II,i} \left( U_i - \frac{\dot{q}_i}{w_{xx,i}} \right), \quad i = \overline{1, N},$$

где  $M_{II,i}$  – пусковые моменты,  $U_i$  – напряжения, подаваемые на двигатели,  $w_{xx,i}$  – скорости холостого хода двигателей.

Некоторые двигатели манипуляционных роботов могут управлять одновременно несколькими звеньями через передаточный механизм. В этом случае на один (основной) шарнир подается момент, а поведение второго (повторяемого) шарнира описывается с помощью ограничения:

$$q_l - kq_m = 0, \quad m, l \in [1, N],$$

где  $q_m$  – координата основного шарнира,  $q_l$  – координата повторяемого шарнира,  $k$  – передаточное число.

Дифференцируя это уравнение, получим ограничение относительно скоростей

$$\dot{q}_l - k\dot{q}_m = 0, \quad m, l \in [1, N].$$

#### Трение и ограничения на параметры в шарнирах.

Повышение адекватности моделирования динамики манипуляционных роботов связано с реализацией моделей трения и ограничений на параметры относительного движения в шарнирах.

Трение в шарнирах бывает сухим и вязким. В импульсном подходе модель сухого трения по Кулону заключается в том, что обобщенный импульс трения в шарнире не превышает своего максимального значения:

$$|p_{fr,i}| \leq p_{\max,i}, \quad i = \overline{1, N}.$$

Это условие дополняется ограничением вида (5) для трения:

$$\dot{q}_i = 0, \quad i = \overline{1, N}.$$

Модель вязкого трения состоит в формировании момента в шарнире:

$$\tau_{fr,i} = -c_i \dot{q}_i, \quad i = \overline{1, N},$$

где  $c_i$  – коэффициент вязкого трения.

Ограничения на параметры в шарнирах описываются в виде

$$q_{\min,i} \leq q_i \leq q_{\max,i}, \quad i = \overline{1, N}.$$

После приведения этого соотношения к виду (4) получим, что должны быть выполнены неравенства  $q_i - q_{\min,i} \geq 0$  и  $q_{\max,i} - q_i \geq 0$ .

Если в процессе моделирования нарушается первое неравенство, то формируется ограничение относительно скоростей  $\dot{q}_i \geq 0$  с вычислением импульса  $p_{\lim} \geq 0$ . В том случае, если нарушается второе неравенство, то  $-\dot{q}_i \geq 0$  с  $p_{\lim} \leq 0$ .

#### Коллизии звеньев системы шарнирно связанных тел.

В процессе моделирования динамики манипуляционных роботов их звенья могут сталкиваться друг с другом, а также с другими объектами виртуального окружения. Это приводит к необходимости решения задачи определения и разрешения коллизий.

На стадии определения коллизий определяется необходимая информация о пересечении объектов: экстремальные точки проникновения геометрий объектов, глубины проникновений и направление, по которому необходимо расталкивать тела. Эта информация далее используется на стадии разрешения коллизий тел для моделирования контакта, удара и различных видов трения между телами.

При наличии пересечения двух звеньев необходимо обеспечить выполнение ограничения следующего вида

$$(\mathbf{r}_p^m - \mathbf{r}_p^l) \mathbf{n} = -d \geq 0, \quad m, l \in [1, N],$$

где  $\mathbf{r}_p^m$  и  $\mathbf{r}_p^l$  – координаты экстремальных точек проникновения двух тел,  $\mathbf{n}$  – направление (нормаль) расталкивания,  $d$  – глубина проникновения.

Если текущая скорость ограничения удовлетворяет неравенству  $(\mathbf{v}_p^m - \mathbf{v}_p^l) \mathbf{n} < -V_{rest}$ , где  $V_{rest}$  – задаваемая скорость покоя, то это соответствует случаю удара и должно быть выполнено ограничение

$$(\mathbf{v}_p^{m,+} - \mathbf{v}_p^{l,+}) \mathbf{n} = -k_{rest} (\mathbf{v}_p^{m,-} - \mathbf{v}_p^{l,-}) \mathbf{n},$$

$$m, l \in [1, N],$$

где  $\mathbf{v}_p^{m,-}$  и  $\mathbf{v}_p^{l,-}$  – скорости тел в точке до удара (вычисляются до применения сил в (1)),  $\mathbf{v}_p^{m,+}$  и  $\mathbf{v}_p^{l,+}$  – скорости тел в точке после удара,  $k_{rest}$  – коэффициент восстановления при ударе.

В противном случае получаем случай контакта двух тел, для которого необходимо обеспечить выполнение ограничения

$$(\mathbf{v}_p^m - \mathbf{v}_p^l)\mathbf{n} \geq 0, \quad m, l \in [1, N].$$

Импульс  $\mathbf{p}_{con} = p_{con}\mathbf{n}$ , обеспечивающий выполнение этого ограничения, должен стремиться расталкивать тела. Поэтому для него должно быть выполнено ограничение на его величину  $p_{con} \geq 0$ .

Моделирование трения между телами осуществляется при наличии их контакта и бывает трех видов: скольжения, качения и верчения. Для этого рассматривается плоскость контакта, определяемая двумя взаимно перпендикулярными векторами  $\mathbf{t}_1$  и  $\mathbf{t}_2$ , которые в свою очередь перпендикулярны вектору  $\mathbf{n}$ . В силу нелинейности модели Кулона для трения скольжения непосредственное приведение к виду (6) невозможно. Поэтому рассматривается подход, в котором конус трения аппроксимируется пирамидой [5]. Это приводит к ограничениям для трения скольжения следующего вида

$$(\mathbf{v}_p^m - \mathbf{v}_p^l)\mathbf{t}_i = 0, \quad |p_{sl,i}| \leq k_{sl}p_{con},$$

$$i = 1, 2, \quad m, l \in [1, N],$$

где  $k_{sl}$  – коэффициент трения скольжения двух материалов тел.

Аналогичным образом формулируются

ограничения для трений качения и верчения между телами:

$$(\boldsymbol{\omega}^m - \boldsymbol{\omega}^l)\mathbf{t} = 0, \quad |p_{rol}| \leq k_{rol}p_{con};$$

$$(\boldsymbol{\omega}^m - \boldsymbol{\omega}^l)\mathbf{n} = 0, \quad |p_{piv}| \leq k_{piv}p_{con},$$

$$m, l \in [1, N],$$

где  $\boldsymbol{\omega}^m$  и  $\boldsymbol{\omega}^l$  – угловые скорости тел,  $\mathbf{t}$  – ось качения тела,  $k_{rol}$  и  $k_{piv}$  – коэффициенты трений качения и верчения для пары материалов тел. В некоторых случаях тело может совершать качение в двух направлениях, что требует обработки двух ограничений для трения качения.

## 4. Результаты моделирования

Предложенные в статье методы и подходы были реализованы в виде программных модулей подсистемы динамики имитационно-тренажерного комплекса, разработанного в ФГУ ФНЦ НИИСИ РАН. В рамках этого комплекса осуществляется имитационное моделирование и управление колесными, гусеничными, антропоморфными роботами и другими объектами виртуального окружения.

На Рис. 1 показана операция захвата радиоактивного контейнера манипулятором гусеничного робота МФ-4. Захватное устройство манипулятора робота представляет собой механизм, в котором за счет введения замкнутой кинематической цепи обеспечивается плоскопараллельное перемещение губок. Кроме того, рассматривается двигатель на схвате, который одновременно управляет двумя губками через

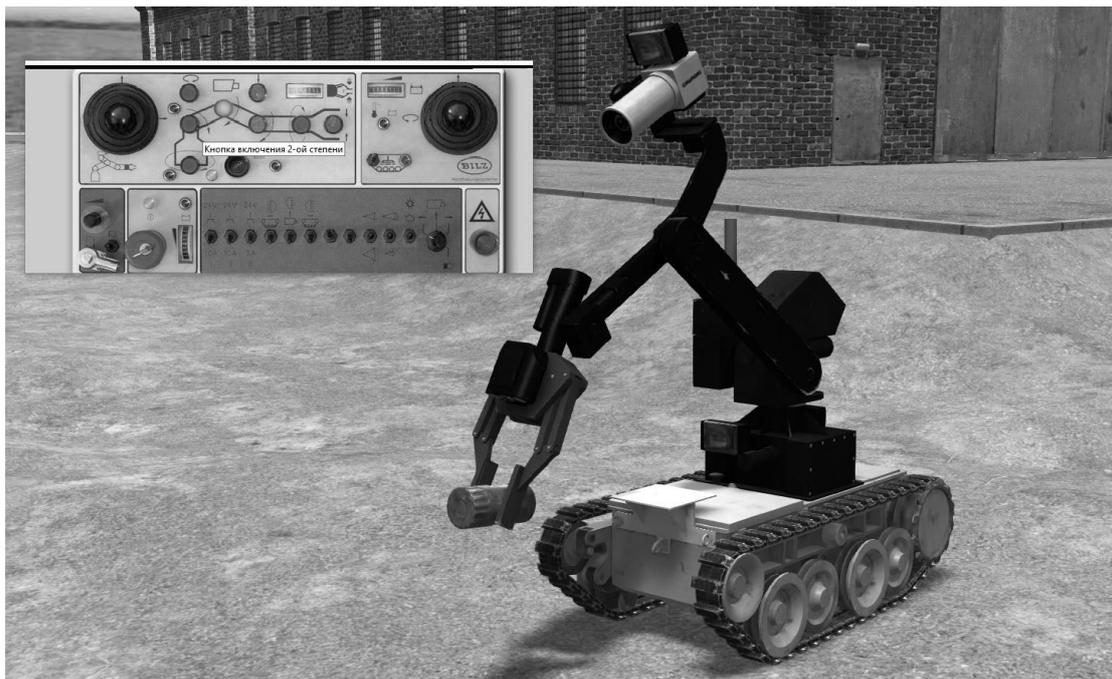


Рис. 1. Захват контейнера роботом МФ-4 с помощью пульта

передаточный механизм. Управление роботом МФ-4 осуществляется в командном режиме путем подачи напряжений для каждого двигателя в сочленениях робота. При захвате объекта реализовано моделирование коллизий губок с объектом с учетом трений скольжения, качения и верчения.

На Рис. 2 приведена модель антропоморфного торсового робота, для которой рассматривается удержание дрели-шуруповерта для выполнения операций с крепежными изделиями. Управление роботом осуществляется в командном и полуавтоматическом режиме. Каждый шарнир робота имеет ограничения на углы поворота, что является приближением биомеханических ограничений человека. Также моделируется сухое и вязкое трение в шарнирах робота.

Применение метода составных тел и его модификаций на основе импульсов позволяет в реальном времени и с высокой точностью

моделировать технологические операции, выполняемые рассматриваемыми манипуляционными роботами.

## Заключение

В работе была рассмотрена задача моделирования динамики манипуляционных роботов в режиме реального времени. Применение метода составных тел и формулировка задачи обработки ограничений в виде MLCР позволяет универсальным и единым образом обрабатывать ограничения различного типа. Предложенный подход позволяет реализовать сложные кинематические механизмы, трение и ограничение на параметры в шарнирах, а также коллизии звеньев манипулятора с другими объектами.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-00137.

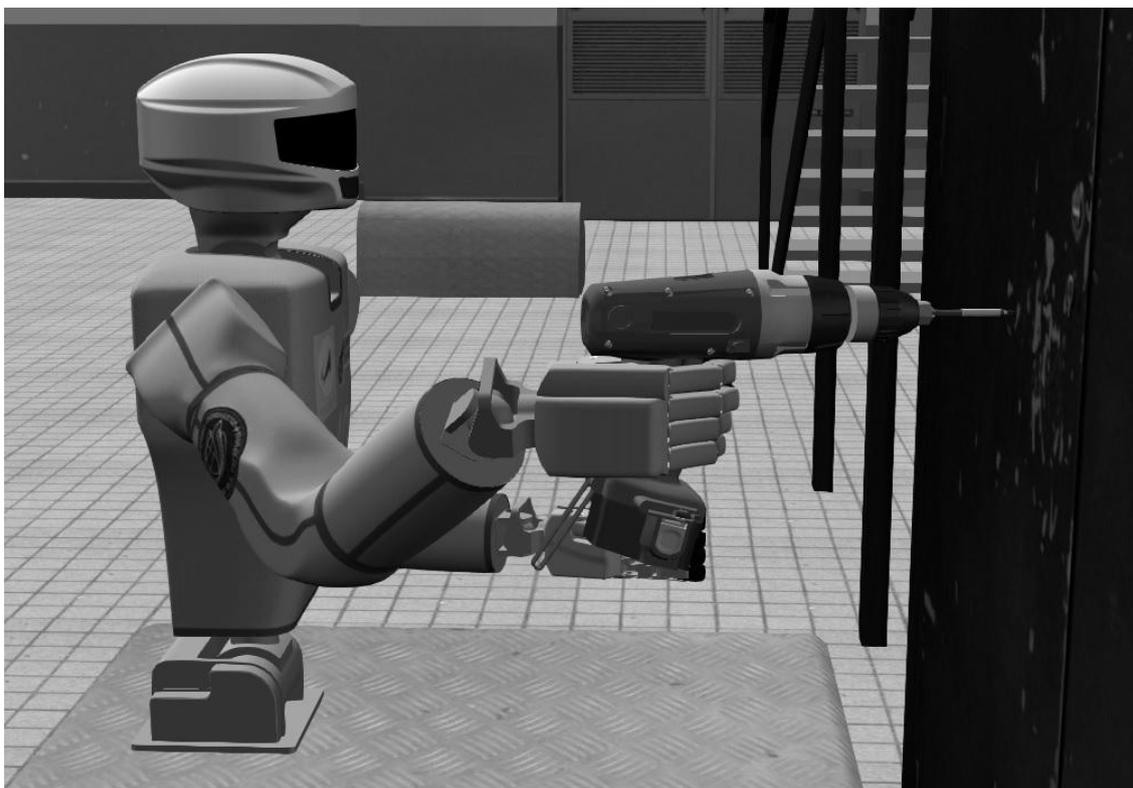


Рис. 2. Торсовый антропоморфный робот

# Dynamic simulation of manipulation robots in virtual environment systems

**E.V. Strashnov, L.A. Finagin**

**Abstract:** This paper considers a problem of real time dynamic simulation of manipulation robots in virtual environment systems. The proposed solution of this problem is to apply the articulated body method and its modifications based on impulses, which allows to realize the dynamics of complex kinematic mechanisms and manipulator end-effectors. In this case, the enforcement of constraints is performed by solving the MLCP (Mixed Linear Complementarity Problem) with respect to the unknown values of the impulses.

**Keywords:** articulated rigid bodies system, real time, articulated body method, impulse-based approach, MLCP, virtual environment systems.

## Литература

1. Е.И. Зенкевич, А.С. Ющенко. Основы управления манипуляционными роботами. 2-е изд. М., Изд-во МГТУ им. Н.Э. Баумана, 2004.
2. R. Featherstone. Rigid body dynamics algorithms. New York, Springer-Verlag, 2008.
3. B. Mirtich. Impulse-based dynamic simulation of rigid body systems. PhD thesis. University of California, Berkeley, 1996.
4. M.V. Mikhaylyuk, E.V. Strashnov and P.Yu. Timokhin. Algorithms of multibody dynamics simulation using articulated-body method // *Mathematica Montisnigri*, 2017, Vol. XXXIX, pp. 133-145.
5. H. Gastenauer. A unified framework for rigid body dynamics. PhD Thesis. Johannes Kepler Universitaet, Linz, Austria, 2006.
6. М.В. Михайлюк, Е.В. Страшнов. Моделирование системы связанных тел методом последовательных импульсов // *Труды НИИСИ РАН*, 2014, т.4, № 2, стр. 52-60.
7. J. Stepien. Physics-based animation of articulated rigid body systems for virtual environments. Gliwice, 2013.
8. А.А. Самарский, А.В. Гулин. Численные методы: Учеб. пособие для вузов. М., Наука, 1989.
9. Е.В. Страшнов, М.А. Торгашев. Моделирование динамики электроприводов виртуальных роботов в имитационно-тренажерных комплексах // *Издательство "Новые технологии"*, Мехатроника, автоматизация, управление, 2016, том 17, № 11, стр. 762-768.

# Эффективные алгоритмы визуализации результатов виртуального моделирования керна с использованием шейдерной тесселяции

П.Ю. Тимохин<sup>1</sup>, М.В. Михайлюк<sup>2</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия, E-mail's: <sup>1</sup>webpismo@yahoo.de, <sup>2</sup>mix@niisi.ras.ru

**Аннотация:** В статье рассмотрена задача визуализации цифровой модели керна, полученной с помощью рентгеновской компьютерной томографии. Предложены новые, эффективные методы и алгоритмы воксельной визуализации выделенного объема цифровой модели керна. Предложенные методы и алгоритмы основаны на распараллеливании вычислений на многоядерном графическом процессоре с помощью шейдерной тесселяции графических примитивов. Созданные методы и алгоритмы были реализованы в программном комплексе «Керноsimулятор», разработанном в ФГУ ФНЦ НИИСИ РАН.

**Ключевые слова:** визуализация, виртуальная модель, керна, рентгеновская компьютерная томография, реальное время, тесселяция, GPU

## Введение

В настоящее время одним из ключевых инструментов для проведения оценки перспективности нефтеносности месторождений трудноизвлекаемых запасов углеводородов является виртуальное моделирование керна - цилиндрического образца породы, извлеченного из пробуренной скважины [1]. В частности, широко востребовано моделирование порового флюида (нефть, газ, вода) и минерального скелета керна с помощью рентгеновской компьютерной томографии (РКТ) [2]. Результатами такого моделирования являются масштабированные значения коэффициента поглощения породой керна рентгеновского излучения в ячейках трехмерной сетки - *цифровая модель керна*. Важной задачей является визуализация в масштабе реального времени (со скоростью не менее 25 кадров в секунду) выделенного объема цифровой модели керна, причем в этом объеме будут визуализироваться только те ячейки, которые соответствуют заданному исследователем диапазону значений коэффициента поглощения.

Эффективным является моделирование и визуализация выделенного объема цифровой модели керна с помощью *вокселей* - аналогов двумерных пикселей для трехмерного пространства. Основными подходами к построению воксельной модели являются

прокладка лучей («ray casting») через пиксели экрана до пересечения с выделенным объемом моделирования [3, 4] и построение полигональной (триангулированной) модели воксельного объема [5, 6]). Первый подход позволяет получать качественные (детализированные) воксельные модели, однако, предъявляет высокие требования к графическим вычислительным ресурсам, которые возрастают с увеличением размеров моделируемого объема и разрешения экрана, что препятствует выполнению синтеза конечных изображений в масштабе реального времени. Второй подход является вычислительно менее затратным, ввиду того, что обработка полигонов в современных графических ускорителях поддерживается на уровне архитектуры графического процессора (GPU). При таком подходе качество воксельной визуализации ограничивается числом полигональных моделей вокселей, которое можно визуализировать в масштабе реального времени. Ввиду этого возникает задача разработки алгоритмов построения на GPU в масштабе реального времени большого количества полигональных моделей вокселей, использующих распараллеливание графических расчетов на современных многоядерных GPU.

В данной работе предлагаются новые алгоритмы решения этой задачи, основанные на использовании современной технологии

шейдерной (программируемой на GPU) тесселяции графических примитивов [7], которые обеспечивают визуализацию в масштабе реального времени освещаемой полигональной модели большого объема (размером до  $10^9$  вокселей). Расчеты в предлагаемых алгоритмах реализуются на шейдерном языке GLSL с использованием графической библиотеки Open GL версии 4.0.

## 1. Полигональная модель воксела

Рассмотрим задачу визуализации некоторого объема  $M$  моделирования из всего массива коэффициентов поглощения в цифровой модели ядра. Пользователь задает этот объем в интерфейсе программного комплекса, указывая отрезки координат  $[x_{begin}, x_{end}] \times [y_{begin}, y_{end}] \times [z_{begin}, z_{end}]$ . Введем обозначения  $n = x_{end} - x_{begin}$ ,  $m = y_{end} - y_{begin}$ ,  $q = z_{end} - z_{begin}$  и обозначим через  $K_{abs}$  трехмерный массив значений коэффициента поглощения в вокселях объема  $M$ . Из объема моделирования мы будем визуализировать только те воксели, в которых коэффициент  $k_{abs} \in [k_{abs, min}, k_{abs, max}]$  (диапазон также задается пользователем).

Обозначим через  $P_e$  позицию наблюдателя в рассматриваемой сцене. Рассмотрим произвольный воксел (кубик) из объема  $M$ . Пронумеруем его вершины, например, как показано на рис. 1. С какой бы стороны наблюдатель ни смотрел на этот воксел, он всегда будет видеть не более трех его граней. Действительно, если воксел расположен ниже и левее точки  $P_e$ , то наблюдатель будет видеть правую, верхнюю и переднюю грани. Такие три грани однозначно задаются своей одной общей вершиной и образуют фигуру, которую мы будем называть *трилистником*. Заметим, что если наблюдатель видит две (или даже одну) грань, то мы добавляем в рассмотрение одну (две) вырожденную грань. Обозначим через Пр, Л, В, Н, П, З – правую, левую, верхнюю, нижнюю, переднюю и заднюю грани кубика. Легко видеть (см. рис. 1), что имеется 8 различных вариантов трилистников: 0 - (Л, Н, З), 1 - (Пр, Н, З), 2 - (Л, В, З), 3 - (Пр, В, З), 4 - (Л, Н, П), 5 - (Пр, Н, П), 6 - (Л, В, П), 7 - (Пр, В, П). Здесь сначала записан номер трилистника, а затем в скобках – образующие его грани.

Чтобы определить, какой вариант трилистника наблюдатель видит из позиции

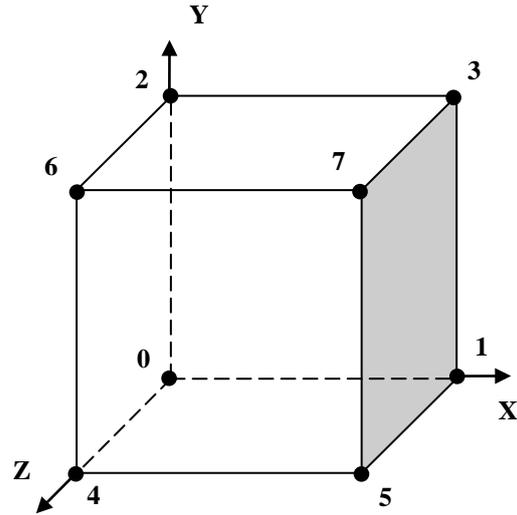


Рис. 1. Номера вершин воксела

$P_e$ , мы в видовой системе координат (View Coordinate System, VCS [8]) проверим значения косинусов углов между вектором  $\vec{e}_{VCS} = 0,0,1$ , обратным направлению взгляда наблюдателя, и нормальными  $\vec{n}_{VCS,0}$ ,  $\vec{n}_{VCS,1}$ ,  $\vec{n}_{VCS,2}$  к правой, верхней и передней граням воксела. Эти нормали будут одинаковы у всех вокселей (для текущего кадра визуализации), и их можно вычислить как

$$\vec{n}_{VCS,p} = M_{NORM} \cdot \vec{n}_{OCS,p}, \quad (1)$$

где  $M_{NORM}$  – матрица преобразования координат нормали в систему VCS (она находится как обратная матрица от транспонированной левой верхней подматрицы  $3 \times 3$  текущей модельно-видовой матрицы [8]), а  $\vec{n}_{OCS,0} = 1,0,0$ ,  $\vec{n}_{OCS,1} = 0,1,0$ ,  $\vec{n}_{OCS,2} = 0,0,1$  – нормали к правой, верхней и передней граням воксела в объектной системе координат (Object Coordinate System, OCS) воксела. Из (1) следует, что  $\vec{n}_{VCS,p}$  будут совпадать со столбцами матрицы  $M_{NORM}$ . Отметим, что после вычисления по формуле (1) нормали  $\vec{n}_{VCS,0}$ ,  $\vec{n}_{VCS,1}$  и  $\vec{n}_{VCS,2}$  должны быть нормализованы.

Косинус угла между вектором  $\vec{e}_{VCS}$  и нормалью  $\vec{n}_{VCS,p}$  будет равен  $\vec{e}_{VCS} \cdot \vec{n}_{VCS,p} = \vec{n}_{VCS,p} \cdot z$ . Тогда номер  $t$  видимого трилистника можно вычислить в соответствии с формулой

$$t = \sum_{p=0}^2 2^p b_p, \quad (2)$$

где  $b_p = 1$ , если  $\vec{n}_{VCS,p} \cdot z \geq 0$ , и  $b_p = 0$  – в противном случае.

Каждый трилистник  $S$  мы будем визуализировать лентой (стрипом) треугольников. Стрип записывается в виде

последовательности номеров вершин, в которой каждые три последовательные вершины образуют треугольник. Для наших трилистников получаем следующие стрипы:

$$\begin{aligned} S_0 &= (5, 4, 1, 0, 3, 2, 2, 0, 6, 4), \\ S_1 &= (4, 0, 5, 1, 7, 3, 3, 1, 2, 0), \\ S_2 &= (1, 0, 3, 2, 7, 6, 6, 2, 4, 0), \\ S_3 &= (5, 1, 7, 3, 6, 2, 2, 3, 0, 1), \\ S_4 &= (1, 5, 0, 4, 2, 6, 6, 4, 7, 5), \\ S_5 &= (0, 1, 4, 5, 6, 7, 7, 5, 3, 1), \\ S_6 &= (0, 4, 2, 6, 3, 7, 7, 6, 5, 4), \\ S_7 &= (4, 5, 6, 7, 2, 3, 3, 7, 1, 5). \end{aligned}$$

В каждом из этих стрипов имеется 2 вырожденных треугольника, которые не будут визуализироваться и потреблять вычислительные ресурсы.

Чтобы визуализация трилистника, заданного стрипом, была правильной (корректно рассчитывалась освещенность граней трилистника в сцене), необходимо для всех трех вершин каждого его треугольника задать одну и ту же нормаль соответствующей этому треугольнику грани кубика. Это нетривиальная задача, ввиду того, что в стрипе каждый следующий треугольник использует две вершины от предыдущего. Для решения этой задачи мы используем технологию «вызывающей вершины» («Provoking Vertex») [9], поддерживаемую современными графическими ускорителями. Ее суть состоит в том, что если в графическом примитиве для некоторого атрибута вершины, например, цвета, отключить режим интерполяции (задать режим закраски «flat-shading»), то на всем графическом примитиве будет действовать цвет одной из его вершин - «вызывающей» (по умолчанию это последняя вершина примитива). В треугольном стрипе «вызывающей» вершиной является каждая  $g + 2$ -ая вершина, где  $g = 1, 2, \dots, 8$  является порядковым номером треугольника в стрипе.

В данной работе мы визуализируем стрипы трилистников с отключенной интерполяцией нормалей их вершин, предварительно задав для «вызывающих» вершин стрипа нормали соответствующих граней кубика. Для этого пронумеруем нормали к граням кубика следующим образом:

- 0 - нормаль  $\vec{n}_{vcs,0}$  к правой грани,
- 1 - нормаль  $\vec{n}_{vcs,1}$  к верхней грани,
- 2 - нормаль  $\vec{n}_{vcs,2}$  к передней грани,
- 3 - нормаль  $-\vec{n}_{vcs,0}$  к левой грани,
- 4 - нормаль  $-\vec{n}_{vcs,1}$  к нижней грани,
- 5 - нормаль  $-\vec{n}_{vcs,2}$  к задней грани.

Для стрипов  $S_0, \dots, S_7$  трилистников получим следующие последовательности номеров нормалей:

$$\begin{aligned} N_0 &= (X, X, 4, 4, 0, 0, 0, 0, 5, 5), \\ N_1 &= (X, X, 5, 5, 1, 1, 1, 1, 3, 3), \\ N_2 &= (X, X, 4, 4, 2, 2, 2, 2, 0, 0), \\ N_3 &= (X, X, 3, 3, 1, 1, 1, 1, 2, 2), \\ N_4 &= (X, X, 4, 4, 5, 5, 5, 5, 3, 3), \\ N_5 &= (X, X, 0, 0, 1, 1, 1, 1, 5, 5), \\ N_6 &= (X, X, 4, 4, 3, 3, 3, 3, 2, 2), \\ N_7 &= (X, X, 2, 2, 1, 1, 1, 1, 0, 0), \end{aligned}$$

где  $X$  – произвольный номер нормали (при обработке «вызывающих» вершин стрипа первые два номера пропускаются).

Для визуализации выделенного объема  $M$  моделирования мы создадим GPU потоки (по одному для каждого вокселя), которые будем называть *воксельными*, и несколько шейдеров, работающих в каждом потоке.

Рассмотрим эти этапы более подробно.

## 2. Вычисление воксельных потоков с помощью шейдерной тесселяции

Визуализация каждого трилистника будет производиться в отдельном потоке GPU. Для создания потоков предлагается использовать механизм программируемой *тесселяции* (разбиения на треугольники) графических примитивов. В качестве примитива мы берем квадрат (патч-квад) и разбиваем его на  $64^2$  маленьких квадрата, которые образуются вновь созданными  $65^2$  вершинами (см. рис. 2).

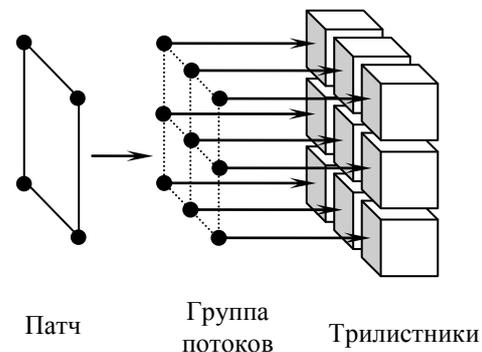


Рис. 2. Генерация трилистников

Для каждой вершины в дальнейшем будет создан GPU поток. Потоки, получившиеся из одного патча, будут образовывать *группу воксельных потоков*. Группы можно записать в одномерный массив размера  $n_p \times m_p \times q_p$ , где  $n_p = \lceil n/65 \rceil$ ,  $m_p = \lceil m/65 \rceil$  и  $q_p = q$ . Таким образом, число необходимых нам патчей также будет равно  $n_p \times m_p \times q_p$ .

Для дальнейшего нам понадобится трехмерный битовый массив  $B[i, j, k]$  размера  $n \times m \times q$ , в котором бит равен 1, если значение

$k_{abs}$  в соответствующем вокселе принадлежит отрезку  $[k_{abs,min}, k_{abs,max}]$ .

Массив патчей в каждом кадре визуализации будет отправляться на графический конвейер GPU и обрабатываться с помощью разработанного комплекса шейдерных программ. Комплекс включает в себя шейдер управления тесселяцией, шейдер вычисления тесселяции, геометрический и фрагментный шейдеры.

### Шейдер управления тесселяцией (УТ).

Данный шейдер работает одновременно на всех ядрах графического процессора и на каждом ядре обрабатывает свой патч. УТ-шейдер вычисляет тройку индексов  $i_p, j_p, k_p$  патча (как будто он записан в трехмерном массиве размера  $n_p \times m_p \times q_p$ ) и числа  $l_w$  и  $l_H$  квадратов, на которые будет разбиваться патч по ширине и высоте в процессе тесселяции. Это реализует следующий алгоритм.

1. Передадим в УТ-шейдер следующие параметры:

- $n_p, m_p, q_p$  - размеры трехмерного массива патчей;
- $n_{slice} = n_p m_p$  - количество двумерных массивов патчей (слоев) в трехмерном массиве патчей;
- $n_{patches} = n_p m_p q_p$  - количество патчей в трехмерном массиве патчей.

2. Вычислим индексы  $i_p, j_p, k_p$  патча:

$$k_p = \left\lfloor \frac{gl\_PrimitiveID}{n_{slice}} \right\rfloor,$$

$$i_p = \left\lfloor \frac{gl\_PrimitiveID - k_p n_{slice}}{n_{patches}} \right\rfloor,$$

$$j_p = gl\_PrimitiveID - k_p n_{slice} - i_p n_p,$$

где  $gl\_PrimitiveID \in [0, n_{patches} - 1]$  - индекс патча в одномерном массиве патчей (присваивается каждому патчу автоматически при поступлении на графический конвейер).

3. Вычислим числа  $l_w$  и  $l_H$  квадратов, на которые будет разбиваться патч по ширине и высоте:

$$l_w = \max \min 64, n - 65j_p - 1, 1,$$

$$l_H = \max \min 64, m - 65i_p - 1, 1.$$

Конец алгоритма.

После выполнения УТ-шейдера генератор примитивов осуществляет разбиение исходного патча на группу вершин (воксельных потоков) размера  $l_w + 1 \times l_H + 1$ . Это фикси-

рованный этап графического конвейера с аппаратно реализованным алгоритмом. Сгенерированные вершины заданы нормализованными вещественными координатами  $u, v$  положения в исходном патче ( $u, v \in [0, 1]$ ).

### Шейдер вычисления тесселяции (ВТ).

После разбиения исходного патча генератором примитивов выполняется ВТ-шейдер, который для каждой вершины из группы сгенерированных вершин (потоков) рассчитывает тройку индексов  $i, j, k$  воксела в трехмерном массиве вокселей. Это реализует следующий алгоритм.

1. Передадим из УТ-шейдера в ВТ-шейдер тройку индексов  $i_p, j_p, k_p$  исходного патча, а также числа  $l_w$  и  $l_H$  квадратов, на которые разбит этот патч по ширине и высоте.

2. Вычислим индексы  $i_v$  и  $j_v$  вершины в группе сгенерированных вершин:

$$i_v = \lfloor l_H v + \varepsilon \rfloor,$$

$$j_v = \lfloor l_w u + \varepsilon \rfloor,$$

где  $u, v$  - нормализованные координаты вершины в исходном патче, а  $\varepsilon = 0.01$  - малая константа, компенсирующая погрешность машинного представления вещественных чисел.

3. Вычислим тройку индексов  $i, j, k$  воксела в трехмерном массиве вокселей:

$$i = 65i_p + i_v,$$

$$j = 65j_p + j_v,$$

$$k = k_p.$$

Конец алгоритма.

**Геометрический шейдер (Г).** Вычисленная в ВТ-шейдере тройка индексов  $i, j, k$  однозначно определяет воксел, который будет визуализироваться в текущем воксельном потоке. Можно считать, что все длины сторон каждого воксела равны единице. Для данного воксела разработанный Г-шейдер выполняет построение полигональной модели (стрипа) трилистника с помощью следующего алгоритма.

1. Передадим в Г-шейдер следующие параметры:

- тройку индексов  $i, j, k$  воксела (из ВТ-шейдера);
- координаты  $n_{VCS,0}, \dots, n_{VCS,5}$  нормалей к граням воксела в системе VCS (см. (1));
- модельно-видовую матрицу  $M_{MV}$ ;
- проекционную матрицу  $M_{PROJ}$ .

2. Если  $B[i, j, k] = 0$ , то  $i, j, k$  -ый воксел не визуализируется, и мы выходим из алгоритма.

3. Запишем координаты  $P_{OCS,0}, \dots, P_{OCS,7}$  всех вершин кубика (воксела) в системе OCS:

$$3.1. \quad x_0 = j, \quad y_0 = i, \quad z_0 = k, \\ x_1 = j+1, \quad y_1 = i+1, \quad z_1 = k+1.$$

$$3.2. \quad P_{OCS,0} = x_0, y_0, z_0, \quad P_{OCS,1} = x_1, y_0, z_0, \\ P_{OCS,2} = x_0, y_1, z_0, \quad P_{OCS,3} = x_1, y_1, z_0, \\ P_{OCS,4} = x_0, y_0, z_1, \quad P_{OCS,5} = x_1, y_0, z_1, \\ P_{OCS,6} = x_0, y_1, z_1, \quad P_{OCS,7} = x_1, y_1, z_1.$$

4. Вычислим координаты  $P_{VCS,0}, \dots, P_{VCS,7}$  и  $P_{CCS,0}, \dots, P_{CCS,7}$  всех вершин кубика в системе VCS и системе координат объема видимости (Clipping Coordinate System, CCS) соответственно:

Цикл по  $v$  от 0 до 7

$$P_{VCS,v} = M_{MV} P_{OCS,v}. \\ P_{CCS,v} = M_{PROJ} P_{VCS,v}.$$

5. Вычислим номер  $t$  видимого трилистника по формуле (2).

6. Зададим координаты  $P'_{CCS,0}, \dots, P'_{CCS,9}$  позиций вершин стрипа  $t$ -го трилистника (см. раздел 1 статьи) в системе CCS, а также координаты  $P'_{VCS,0}, \dots, P'_{VCS,9}$  позиций и координаты  $n'_{VCS,0}, \dots, n'_{VCS,9}$  нормалей вершин этого стрипа в системе VCS (необходимы для последующих расчетов во фрагментом шейдера):

Цикл по  $s$  от 0 до 9

Запишем номер  $a = N_t, s$ , где  $N_t$  - последовательность номеров нормалей для вершин стрипа  $t$ -го трилистника (см. раздел 1).

Зададим  $n'_{VCS,s} = n_{VCS,a}$ .

Запишем номер  $b = S_t, s$ , где  $S_t$  - последовательность индексов вершин кубика для стрипа  $t$ -го трилистника (см. раздел 1).

Зададим  $P'_{CCS,s} = P_{CCS,b}$ .

Зададим  $P'_{VCS,s} = P_{VCS,b}$ .

Конец цикла.

8. Построим стрип  $t$ -го трилистника по заданным позициям  $P'_{CCS,0}, \dots, P'_{CCS,9}$  вершин.

Конец алгоритма.

Отметим, что в п. 7 алгоритма задание координат  $n'_{VCS,0}, \dots, n'_{VCS,9}$  нормалей и дополнительных координат  $P'_{VCS,0}, \dots, P'_{VCS,9}$  позиций вершин в системе VCS позволяет

упростить последующие расчеты во фрагментном шейдере.

**Фрагментный шейдер (Ф).** Полученный из Г-шейдера стрип трилистника проходит растеризацию (фиксированный этап графического конвейера), на которой он разбивается на пиксели (фрагменты), образующие изображение трилистника. Вычисление цветов пикселей выполняется параллельно, независимо друг от друга с помощью разработанного Ф-шейдера на основе модели затенения Фонга [8] (мы используем направленный бесконечно удаленный источник освещения). Это реализует следующий алгоритм.

1. Передадим в Ф-шейдер следующие параметры:

- координаты нормали  $\vec{n}_{VCS}$  к поверхности трилистника в рассматриваемой точке в системе VCS (из пункта 6 алгоритма Г-шейдера, одно из значений  $n_0, \dots, n_9$ );

- коэффициенты  $K_A = k_{A,r}, k_{A,g}, k_{A,b}, k_{A,a}$ ,  $K_D = k_{D,r}, k_{D,g}, k_{D,b}, k_{D,a}$ ,  $K_S = k_{S,r}, k_{S,g}, k_{S,b}, k_{S,a}$

фонового, диффузного и зеркального отражения материала модели воксела;

- коэффициент  $k_{shine}$  зеркального блеска материала модели воксела;

- интенсивности  $I_A = r_A, g_A, b_A, a_A$ ,  $I_D = r_D, g_D, b_D, a_D$ ,  $I_S = r_S, g_S, b_S, a_S$  фоновой, диффузной и зеркальной составляющих освещения;

- координаты вектора  $\vec{l}_{VCS}$  на источник освещения, в системе VCS;

- координаты  $P_{VCS,v}$  рассматриваемой точки поверхности трилистника, в системе VCS (вычисляются автоматически графическим конвейером из координат  $P'_{VCS,0}, \dots, P'_{VCS,9}$ ).

2. Вычислим диффузную составляющую  $C_D$  цвета пикселя:

$$C_D = K_D I_D \max \vec{n}_{VCS}, \vec{l}_{VCS}, 0.$$

3. Вычислим зеркальную составляющую  $C_S$  цвета пикселя:

Запишем координаты вектора  $\vec{v}_{VCS}$  направления из рассматриваемой точки поверхности трилистника на наблюдателя, в системе VCS:

$$\vec{v}_{VCS} = 0, 0, 0 - P_{VCS,v} = -P_{VCS,v},$$

где  $0, 0, 0$  - координаты позиции наблюдателя в системе VCS.

Вычислим координаты вектора  $\vec{r}_{VCS}$  отраженного луча света в рассматриваемой точке поверхности трилистника:

$$\vec{r}_{VCS} = 2 \vec{n}_{VCS}, \vec{l}_{VCS} - \vec{n}_{VCS} - \vec{l}_{VCS}.$$

Нормализуем вектор  $\vec{r}_{VCS}$ .

Вычислим зеркальную составляющую

$$C_S = K_S I_S \max \{ \vec{v}_{VCS}, \vec{r}_{VCS}, 0 \}^{k_{shine}}.$$

4. Вычислим результирующий цвет  $C$  пиксела:

$$C = K_A I_A + C_D + C_S.$$

Конец алгоритма.

Вычислив цвета всех пикселей изображений всех трилистников, мы получаем результирующее изображение выделенного объема  $M$  моделирования.

### 3. Результаты

На основе предложенных методов и алгоритмов был разработан программный комплекс визуализации в масштабе реального времени результатов виртуального моделирования ядра. Для апробации комплекса использовались цифровые модели образцов кернов баженовской свиты и песчаника (размеры образцов -  $2560 \times 2560 \times 8600$  и  $2600 \times 2600 \times 7360$  вокселей соответственно). Была выполнена визуализация выделенного объема цифровой модели ядра размером  $400^3$ ,  $600^3$  и  $1000^3$  вокселей для различных диапазонов значений коэффициента поглощения, соответствующих поровому пространству ядра (см. рис. 3). Визуализация

### Заключение

В данной работе рассмотрена задача визуализации цифровой модели ядра, полученной с помощью рентгеновской компьютерной томографии. Было предложено решение, обеспечивающее воксельную визуализацию в масштабе реального времени выделенного объема цифровой модели ядра с учетом задаваемого исследователем диапазона значений коэффициента поглощения. Предложенное решение включает эффективные методы моделирования и визуализации воксельного объема с помощью полигональных «трилистников» и воксельных потоков, а также новые, эффективные алгоритмы создания и вычисления воксельных потоков на ядрах GPU с помощью разработанных шейдеров.

Предложенные методы и алгоритмы были реализованы в программном комплексе «Керноsimулятор», разработанном в ФГУ ФНЦ НИИСИ РАН. Была выполнена апробация комплекса на цифровых моделях образцов кернов баженовской свиты и песчаника, которая подтвердила адекватность предложенных решений поставленной задаче. Разработанные решения могут быть также использованы при построении систем научной визуализации, систем виртуального окружения, виртуальных лабораторий и др.

Публикация выполнена в рамках государственного задания по проведению

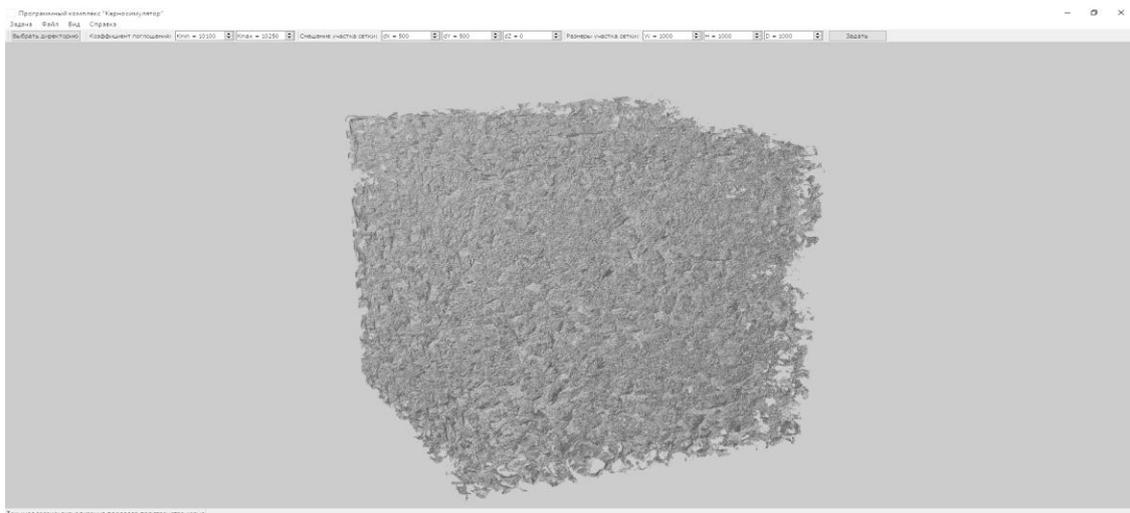


Рис. 3. Пример кадра визуализации участка цифровой модели ядра размером  $1000^3$  вокселей

выполнялась при разрешении Full HD на персональном компьютере, оборудованном видеокартой NVIDIA GeForce GTX 1080 Ti (3584 вычислительных ядра) со скоростью не менее 25 кадров в секунду.

фундаментальных научных исследований (ГП 14) по теме (проекту) «34.9. Системы виртуального окружения: технологии, методы и алгоритмы математического моделирования и визуализации» (0065-2018-0020).

# Effective algorithms to visualize the results of virtual simulation of the core material using shader tessellation

P.Yu. Timokhin, M.V. Mikhaylyuk

**Abstract:** The paper deals with the task of visualization of a digital model of the core material obtained via X-ray computed tomography. New, effective methods and algorithms of voxel visualization of the selected volume of the digital model are proposed. The methods and algorithms proposed are based on parallelization of computations on a multi-core GPU using shader tessellation of graphics primitives. The methods and algorithms created were implemented in the "Core Material Simulator" program complex developed in SRISA RAS.

**Keywords:** visualization, virtual model, core material, X-ray computed tomography, real-time, tessellation, GPU.

## Литература

1. В.Б. Бетелин. Цифровое месторождение – путь к трудноизвлекаемым запасам углеводородов // Инновации. – 2014. – № 1. – С. 37-38.
2. В.Б. Бетелин, В.Ф. Никитин, Н.Н. Смирнов, Е.В. Михальченко, Е.И. Скрылева, Л.И. Стамов, В.В. Тюренкова. Компьютерный керносимулятор: подходы и методы // Вестник кибернетики. – 2015. – № 4. – С. 33-44.
3. A. Kaufman, D. Cohen, R. Yagel, Volume graphics // Computer. – 1993. – Vol. 26, № 7. - PP. 51–64.
4. C. Crassin, F. Neyret, S. Lefebvre, and E. Eisemann, Gigavoxels: Ray-guided streaming for efficient and detailed voxel rendering // Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games, ACM, 2009, PP. 15-22.
5. М.В. Михайлюк, Д.А. Кононов, Д.М. Логинов. Визуализация порового пространства в цифровой модели керна // ИТ-стандарт. – 2017. – № 3-1 (12). – С. 7-10.
6. OpenGL Minecraft Style Volume Rendering. URL: <http://bytebash.com/2012/03/opengl-volume-rendering/> (дата обращения: 26.10.2018).
7. M.Bailey, S.Cunningham. Graphics Shaders: Theory and Practice, Second Edition // CRC Press. – 2011. – 518 p.
8. R.J.Rost, V.Licea-Kane. OpenGL Shading Language (3rd Edition). – Boston, USA: Addison Wesley Professional, 2009. – P. 792.
9. Primitive. Provoking Vertex. URL: [https://www.khronos.org/opengl/wiki/Primitive#Provoking\\_vertex](https://www.khronos.org/opengl/wiki/Primitive#Provoking_vertex) (дата обращения: 26.10.2018).

# Методы обеспечения отвода тепла в корпусах высокопроизводительных защищенных серверов с воздушным охлаждением

А.В. Бакалдин<sup>1</sup>, П.И. Кандалов<sup>2</sup>, Н.А. Козлов<sup>3</sup>, А.Г. Мадера<sup>4</sup>, О.В. Сердин<sup>5</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

E-mail: <sup>1</sup>[bakaldin@cs.niisi.ras.ru](mailto:bakaldin@cs.niisi.ras.ru), <sup>2</sup>[petrki87@gmail.com](mailto:petrki87@gmail.com),

<sup>3</sup>[kozlov\\_n@cs.niisi.ras.ru](mailto:kozlov_n@cs.niisi.ras.ru), <sup>4</sup>[omm@niisi.ras.ru](mailto:omm@niisi.ras.ru), <sup>5</sup>[serdin@cs.niisi.ras.ru](mailto:serdin@cs.niisi.ras.ru)

**Аннотация:** Передача, накопление, анализ и распределение больших массивов данных предъявляет повышенные требования к скорости их обработки, полосе пропускания сигналов, устойчивости и помехозащищенности. Создаваемые для решения этих задач высокопроизводительные компьютерные системы по спецификациям VPX (VITA 46) и repVPX (VITA 65), наряду с высокой пропускной способностью и помехоустойчивостью, обладают также повышенным энергопотреблением. Это приводит к значительному тепловыделению, являющимся препятствием на пути создания эффективных и конкурентоспособных систем. В настоящей работе рассмотрены методы обеспечения отвода тепла в корпусах высокопроизводительных защищенных серверов с принудительным воздушным охлаждением. Приведены конкретные конструкции систем теплоотвода, методики теплового проектирования и полученные результаты.

**Ключевые слова:** массивы данных, высокопроизводительные вычислительные системы, тепловые режимы, защищенный сервер, отвод тепла, микропроцессор, воздушное охлаждение

## 1. Введение

Современные радиолокационные установки, эхолоты, системы обработки изображений генерируют большие потоки данных, которые необходимо накапливать, анализировать и распределять по потребителям без единой ошибки. Авиационные приложения также отличаются значительными объемами информации, пользуясь которой оператор становится более осведомленным относительно текущей обстановки и может принимать более обоснованные решения. Необходимость массовой передачи данных по информационным сетям порождает потребность в более высокой производительности обработки и широкой полосе пропускания каналов связи.

Для решения этих задач наиболее подходят высокопроизводительные модульные компьютерные системы (КС), разработанные в соответствии со спецификациями VPX (VITA 46) и OpenVPX (VITA 65).

Отличительными характеристиками этих КС являются:

- высокая пропускная способность объединительной панели (используется интерфейс SRIO со скоростью передачи данных до 6.25 Гбод на одну дифференциальную пару контактов);

- высокие параметры нечувствительности к электромагнитным помехам;
- реализованная концепция «заменяем только модуль, а не блок целиком» при ремонте оборудования в полевых условиях неквалифицированным персоналом;
- высокое энергопотребление электронных модулей (максимальное потребление модуля при напряжении питания 5 В постоянного тока составляет 115 Вт, при 12 В – 384 Вт);
- предназначены для использования в жестких условиях эксплуатации.

Суммарная потребляемая мощность таких высокопроизводительных модульных КС может достигать нескольких киловатт. При этом температура окружающей среды, в которой должны функционировать КС, варьируется в диапазоне  $-60^{\circ}\text{C} \div +55^{\circ}\text{C}$ . В этих условиях для обеспечения необходимого теплового режима компонентов, входящих в состав КС, требуются новые подходы для эффективного отвода тепла [1, 2].

В настоящей работе рассмотрены методы обеспечения отвода тепла в корпусах высокопроизводительных защищенных серверов с воздушным охлаждением для ЭВМ, в состав которой входит 16 электронных модулей и два источника питания общей мощностью 1200 Вт, размещенных в каркасе Евромеханика 6U с объединительной панелью.

## 2. Принципы охлаждения ЭВМ и исходные данные для тепловых расчетов

ЭВМ представляет собой принудительно продуваемую конструкцию, в посадочных местах которой устанавливаются 16 электронных модулей (ЭМ), и два источника электропитания (ИЭП) (рис. 1). Охлаждение ЭМ и ИЭП соответствует принудительному обдуву воздухом, с температурой на входе в ЭВМ не более 50°C.

Мощность потребления каждого ЭМ

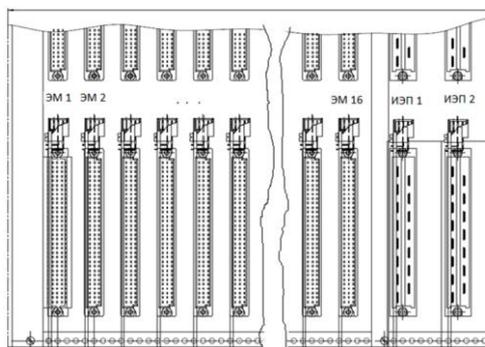


Рис. 1. Корпус ЭВМ с посадочными местами для ЭМ и ИЭП

составляет  $P_{ЭМ} = 60$  Вт, мощность потребления каждого ИЭП равна  $P_{ИЭП} = 120$  Вт, суммарная потребляемая мощность ЭВМ составляет:

$$P = 16 \cdot P_{ЭМ} + 2 \cdot P_{ИЭП} = 1200 \text{ Вт.}$$

Конструкция ЭМ содержит две микросхемы (МС) микропроцессора с архитектурой КОМДИВ64 с мощностью потребления 25 Вт каждая. Размер контактной

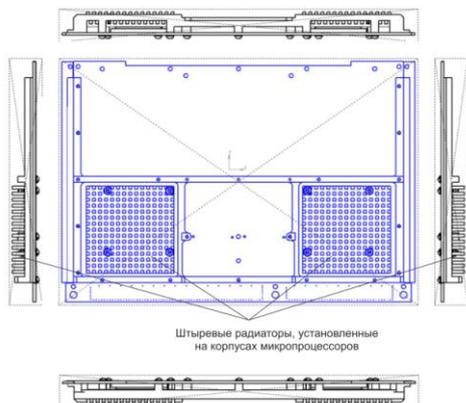


Рис. 2. ЭМ с микропроцессорами, на корпусах которых установлены штыревые радиаторы

площадки для тепловода МС микропроцессора 30 мм×30 мм. Шаг установки ЭМ равен 20,32 мм, глубина

корпуса 280 мм, габаритные размеры корпуса – длина 484 мм, высота 340 мм, шаг установки ИЭП равен 40,64 мм.

Конструкция корпуса ЭВМ не содержит никаких мощных, металлоемких направляющих, пазов и прочих элементов, которые могли бы служить теплоотводами от ЭМ, поэтому основным принципом охлаждения ЭМ и ИЭП является принудительная конвекция с поверхностями МС и ЭРЭ в продуваемый через промежутки между ЭМ и ИЭП воздух. В этом случае, при охлаждении МС процессора, обладающих значительным тепловыделением, для интенсификации отвода тепла от МС микропроцессора, необходимо установить теплообменники в виде радиаторов с развитыми поверхностями теплообмена, наиболее эффективными из которых являются

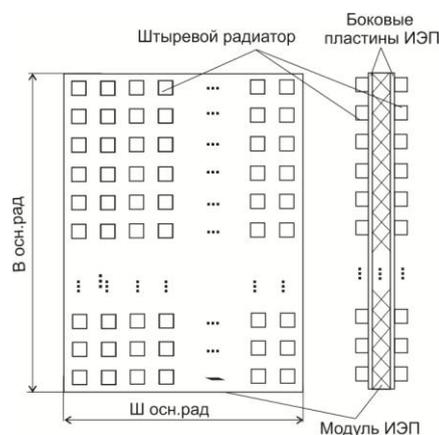


Рис. 3. Модуль ИЭП, с выполненными на его боковых пластинах штыревыми радиаторами

штыревые радиаторы (рис. 2). Штыревые радиаторы крепятся к корпусу МС микропроцессора, к его крышке. Материал штыревого радиатора – медь.

Ввиду значительной величины мощности, потребляемой ИЭП ( $P_{ИЭП} = 120$  Вт), а также принятому в ЭВМ способу охлаждения (принудительной конвекцией в воздух), для интенсификации отвода тепла от ИЭП, на внешних поверхностях боковых пластин ИЭП, ограничивающих с двух сторон модуль ИЭП, необходимо установить, или выполнить, штыревые радиаторы.

Здесь возможно два варианта:

Вариант 1. Штыревые радиаторы крепятся на внешних поверхностях боковых пластин конструкции ИЭП (рис. 3). В этом случае в месте контакта основания радиатора и боковой пластины ИЭП возникает значительное контактное тепловое сопротивление, которое достаточно сложно уменьшить до приемлемой величины ввиду большой площади контакта (даже с

применением теплопроводящих паст и прокладок в месте контакта).

Вариант 2. Штыревые радиаторы выполняются непосредственно как боковые пластины ИЭП (как единое изделие), что позволяет, во-первых, устранить наличие нежелательного контактного теплового сопротивления (см. вариант 1) и, следовательно, снизить температуру ИЭП, и, во-вторых, уменьшить толщину конструкции ИЭП, по сравнению с вариантом 1.

Далее принят вариант 2 конструкции ИЭП, в котором обе боковые пластины ИЭП выполнены в виде штыревых радиаторов.

Конструкция штыревых радиаторов и боковых пластин для модулей ИЭМ корпуса ЭВМ необходимо выполнить из сплава алюминия с высоким значением теплопроводности. Возможные для применения в конструкции корпуса ЭВМ алюминиевые сплавы приведены в таблице 1 [3]. Использование алюминиевого сплава АМг6, в силу его низкой теплопроводности (90 Вт/(м·К)), является неприемлемым.

Таблица 1. Теплопроводность алюминиевых сплавов рекомендуемых к применению

Алюминиевый сплав	Теплопроводность при температуре 300 К, Вт/(м·К)
АВ	176
АД1 нагартованный	210
АД31 закаленный, искусственно состаренный	190
АД35	170
АК6 закаленный, искусственно состаренный	176
АМг1	184
АМц нагартованный	180

### 3. Конструкции радиаторов на корпусе микропроцессора в ЭМ и корпусе ИЭП в ЭВМ

В данном разделе приведены конструкции штыревых радиаторов, устанавливаемых на корпусах МС микропроцессора (рис. 2), и штыревых радиаторов, выполненных на боковых пластинах конструкции корпуса ИЭП (рис. 3).

#### 3.1 Конструкция штыревого радиатора, на корпусе микропроцессора

Проектируемый штыревой радиатор прижимается к корпусу (крышке корпуса) микропроцессора (рис. 2), при этом контакт между корпусом и радиатором осуществляется через теплопроводную прокладку (например, фольгу ТРГ) или теплопроводную пасту (например, Arctic Silver 5).

Примем следующие размеры штыревого радиатора:

- материал штыревого радиатора медь с коэффициентом теплопроводности  $\lambda_{\text{рад}} = 395$  Вт/(м·К);
- основание радиатора: квадрат со стороной  $l = 58$  мм;
- толщина основания радиатора  $d = 3$  мм;
- сечение штыря: квадрат, со стороной штыря  $\delta_{\text{шт}} = a = 2$  мм;
- высота штыря  $h = 8$  мм;
- расстояние между штырями  $b = 2$  мм.

Тогда получим следующие характеристики штыревого радиатора:

- площадь сечения штыря  $S_{1,\text{шт}} = a^2 = 4$  мм<sup>2</sup>;
- периметр сечения штыря  $p_{1,\text{шт}} = 8$  мм;
- количество штырей радиатора  $N$ , размещаемых на прямоугольном основании радиатора с размерами  $l = 58$  мм ( $\delta_{\text{шт}} = a$ )

$$N = \frac{(l+a)(l+a)}{4a^2} = \frac{(58+2)(58+2)}{4 \cdot 2^2} = 225 \text{ шт.}$$

при этом по каждой стороне укладывается по 15 штырей.

#### 3.2 Конструкция штыревых радиаторов на боковых пластинах конструкции ИЭП

Проектируемый штыревой радиатор выполняется на каждой боковой пластине ИЭП (рис. 3) и имеет следующие размеры:

- материал штыревого радиатора (а также боковых пластин ИЭП) – алюминиевый сплав с коэффициентом теплопроводности ( $\lambda_{\text{рад}}$ ) не менее 130 Вт/мК (далее в расчетах радиатора принято  $\lambda_{\text{рад}} = 130$  Вт/(м·К));
- вертикальный и горизонтальный размеры боковых пластин модуля ИЭП равны 233 мм и 160 мм, соответственно. Примем, что штыри радиатора размещены на основании радиатора боковых пластин ИЭП с вертикальными и горизонтальными размерами (по направлению движения потока воздуха)  $V_{\text{осн.рад}} = 223$  мм,  $Ш_{\text{осн.рад}} = 150$  мм;
- толщина основания радиатора (боковой пластины ИЭП)  $d = 2$  мм;
- сечение штыря: квадрат, со стороной штыря  $\delta_{\text{шт}} = a = 3$  мм;
- высота штыря  $h = 4$  мм;
- расстояние между штырями  $b = 5$  мм.

Получим следующие характеристики штыревого радиатора:

- площадь сечения штыря  $S_{1,шт} = a^2 = 9 \text{ мм}^2$ ;
- периметр сечения штыря  $p_{1,шт} = 12 \text{ мм}$ ;
- количество штырей радиатора  $N$ , размещаемых на прямоугольном основании радиатора ( $\delta_{шт} = a$ ) равно:

$$N = \frac{(V_{осн,рад} + b)(Ш_{осн,рад} + b)}{(a+b)^2} \approx 552 \text{ шт.}$$

Примем число штырей радиатора равным 551 шт., при котором по сторонам  $V_{осн,рад} = 223 \text{ мм}$ ,  $Ш_{осн,рад} = 150 \text{ мм}$  укладывается целое число штырей 29 и 19, соответственно.

#### 4. Результаты расчета принудительного конвективного теплообмена в ЭВМ

Мощности, потребляемые ЭМ и ИЭП и преобразуемые в теплоту, частично поглощаются потоком воздуха, проходящим через каналы между ЭМ и ИЭП, и уносятся затем с потоком воздуха за пределы ЭВМ. Количество поглощаемого потоком воздуха тепла определяется теплоемкостью и массой воздуха, а также аэродинамическими характеристиками потока и интенсивностью теплообмена между воздухом и тепловыделяющими ЭМ и ИЭП.

Микросхемы микропроцессоров, установленных в ЭМ, потребляют значительную мощность ( $P_{ЭМ} = 25 \text{ Вт}$ ), поэтому на корпусах МС устанавливаются штыревые радиаторы (рис. 2), поскольку аэродинамические характеристики потока воздуха, проходящего непосредственно через штыревой радиатор, также как и интенсивность теплообмена воздуха со штырями радиатора, существенно превосходят по своей интенсивности аэродинамические и теплообменные характеристики остальных элементов и участков поверхности ЭМ, равно как и иных конструкций радиаторов. При расчетах теплового режима ЭМ, в конструкции которого применяются штыревые радиаторы, необходимо принимать во внимание значительный разброс параметров аэродинамических и тепловых режимов по площади ЭМ, относительно усредненных параметров, при этом максимальные значения разброса будут в области размещения микросхем со штыревыми радиаторами.

Источники электропитания ЭВМ потребляют значительную мощность ( $P_{ИЭП} = 120 \text{ Вт}$ ), которая преобразуется в теплоту и

уносится потоком воздуха, проходящим через ИЭП и каналы, между двумя ИЭП и между ИЭП и ЭМ. Для эффективного отвода тепла от ИЭП указанной мощности, также требуется установка штыревых радиаторов. В этом случае, штыревые радиаторы устанавливаются с двух сторон корпуса ИЭП и выполняются в виде единой конструкции с боковыми пластинами (рис. 3), имеющие размеры, приведенные в п. 3.2.

Ввиду локальных различий параметров аэродинамических и тепловых режимов между областями с штыревыми радиаторами и областями, в которых они отсутствуют, расчеты аэродинамических и тепловых режимов ЭВМ проводятся по методике, содержащей следующие этапы:

Этап 1. Определяются параметры тепловой модели ЭВМ (рис. 4),

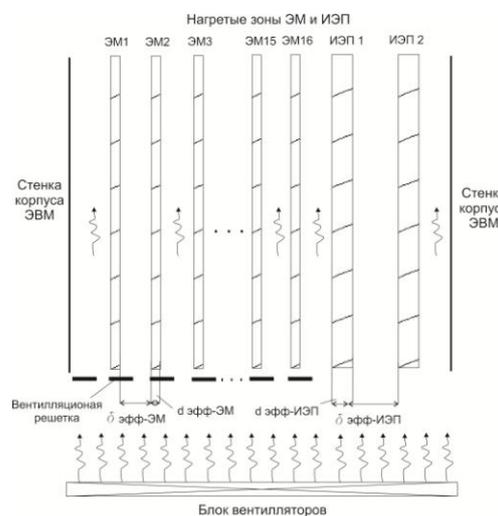


Рис. 4. Тепловая расчетная модель ЭВМ

представляющей собой совокупность 18-ти «нагретых зон» — прямоугольных параллелепипедов, которые моделируют теплообмен и аэродинамику 16-ти ЭМ и 2-х ИЭП с общей мощностью потребления  $P = 1200 \text{ Вт}$  и параметры аэродинамического и теплового режимов ЭВМ:

- размеры нагретых зон ЭМ и ИЭП, учитывающие реальное расположение и размеры элементов в ЭМ и ИЭП, включая спроектированные штыревые радиаторы;
- ширина (усредненная) каналов потока воздуха между нагретыми зонами;
- параметры аэродинамического и теплового режимов потока воздуха — скорость воздуха, средняя температура воздуха в каналах, температура воздуха на выходе из каналов, коэффициенты теплоотдачи.

Этап 2. Расчет аэродинамических и тепловых режимов 16-ти ЭМ, в составе которых находятся оставшиеся МС и ЭРЭ, за

исключением 2-х процессоров на каждом ЭМ. При этом мощность потребления ЭМ равна  $P_{ЭМ-ост} = P_{ЭМ} - 2 \cdot P_{МС-проц} = 10$  Вт.

Этап 3. Расчет аэродинамических и тепловых режимов процессоров в ЭМ, потребляющих мощность  $P_{МС-проц} = 25$  Вт, на корпусах которых установлены штыревые радиаторы.

Этап 4. Расчет аэродинамических и тепловых режимов ИЭП, потребляющих мощность  $P_{ИЭП} = 120$  Вт, со штыревыми радиаторами на двух боковых пластинах корпуса ИЭП.

Результаты расчетов по приведенной методике приведены ниже.

#### 4.1 Расчет параметров тепловой модели теплового режима ЭВМ

##### Эффективная ширина каналов между ЭМ и ИЭП.

Конструкция ЭМ включает в себя множество элементов (МС, ЭРЭ, теплоотводы, мезонины, элементы крепления, соединители) расположенных с двух сторон платы и различающихся между собой высотой и площадью, занимаемой на плате, поэтому канал, образованный двумя соседними ЭМ имеет неравномерную ширину в направлении движения воздушного потока. Эффективная (усредненная) ширина канала  $\delta_{эфф-ЭМ}$  между соседними ЭМ определяется методом нагретой зоны [4, 5, 6] по формуле:

$$\delta_{эфф-ЭМ} = H_{ЭМ} - d_{эфф-ЭМ} = 9,4 \text{ мм},$$

где  $H_{ЭМ} = 20,32$  мм – шаг между соседними ЭМ;  $d_{эфф-ЭМ}$  – ширина нагретой зоны ЭМ, определяемая по формуле  $d_{эфф-ЭМ} = \frac{V_{элемент-ЭМ}}{S_{нз-ЭМ}} = 10,9$  мм, в которой  $V_{элемент-ЭМ}$  – суммарный объем элементов (МС, ЭРЭ, теплоотводы, мезонины, элементы крепления, соединители), расположенных на плате ЭМ с двух сторон;  $V_{элемент-ЭМ} = 383188$  мм<sup>3</sup>;  $S_{нз-ЭМ}$  – площадь нагретой зоны, равная в данном случае, размерам ЭМ в направлении движения потока ( $L_{1,ЭМ} = 150$  мм,  $L_{2,ЭМ} = 233$  мм) воздуха;  $S_{нз-ЭМ} = 34950$  мм<sup>2</sup>.

Для принятой здесь конструкции ИЭП, две его боковые пластины выполнены в виде штыревых радиаторов. Эффективная ширина канала  $\delta_{эфф-ИЭП}$  между двумя ИЭП, определяемая по методу нагретой зоны (см. выше), равна:

$$\delta_{эфф-ИЭП} = H_{ИЭП} - d_{эфф-ИЭП} = 7,24 \text{ мм},$$

где  $H_{ИЭП} = 40,64$  мм – шаг между соседними ИЭП;  $d_{эфф-ИЭП}$  – ширина нагретой зоны ИЭП, определяемая по формуле  $d_{эфф-ИЭП} = \frac{V_{элемент-ИЭП}}{S_{нз-ИЭП}} = 33,4$  мм, в которой  $V_{элемент-ИЭП}$  – суммарный объем ИЭП, включая два

радиатора по его боковым пластинам;  $V_{элемент-ИЭП} = 0,001168$  м<sup>3</sup>;  $S_{нз-ИЭП}$  – площадь нагретой зоны ИЭП, равная в данном случае его размерам в направлении движения ( $L_{1,ИЭП} = 150$  мм,  $L_{2,ИЭП} = 233$  мм) потока воздуха;  $S_{нз-ИЭП} = 34950$  мм<sup>2</sup>.

Эффективная (усредненная) ширина каналов между всеми ЭМ, ИЭП и боковыми стенками корпуса ЭВМ, составит

$$\delta_{эфф} = \frac{16 \cdot 9,4 + 3 \cdot 7,24}{19} = 9,1 \text{ мм}.$$

##### Расчет параметров теплового и аэродинамического режимов потока воздуха в ЭВМ.

Уравнения теплового баланса, описывающие теплообмен в конструкции ЭВМ имеют вид [4, 5, 6]:

$$P = c \cdot G_p (T_{в.вх} - T_{в.вых}),$$

$$T_{в} = 0,5 (T_{в.вх} + T_{в.вых}),$$

где  $P = 1200$  Вт – суммарная мощность потребления ЭВМ;  $G_p$ , кг/м<sup>3</sup> – массовый расход воздуха, поступающий на вход ЭВМ;  $T_{в.вх} = 50^\circ\text{C}$ ,  $T_{в.вых}$  – температуры воздуха на входе и выходе каналов;  $c$  – удельная теплоемкость воздуха;  $\alpha_{кан}$  – коэффициент теплоотдачи от нагретых зон ЭМ и ИЭП к нагнетаемому по каналам воздушному потоку;  $T_{в}$  – средняя температура воздуха в каналах ЭВМ.

Коэффициент теплоотдачи ( $\alpha_{кан}$ ) с

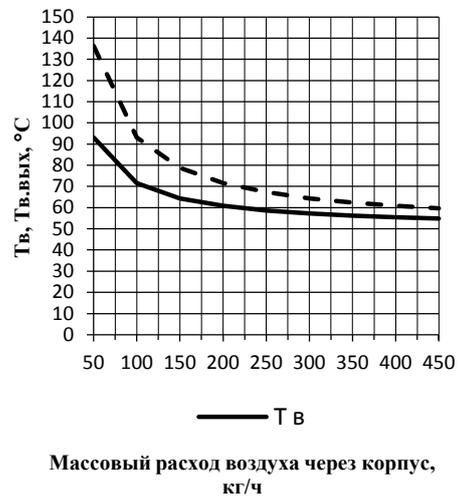


Рис. 5. Зависимости средней температуры воздуха  $T_{в}$  в каналах между ЭМ и ИЭП и температуры воздуха на выходе из каналов  $T_{в.вых}$  в ЭВМ от массового расхода воздуха через корпус

поверхности нагретых зон ЭМ и ИЭП в воздух, нагнетаемый по каналам, определяется критерием Нуссельта

$$\alpha_{кан} = Nu \frac{\lambda}{\delta_{эфф}},$$

зависящим от критерия Рейнольдса [5, 6]

$$Re = \frac{v_{\text{кан}} \cdot \delta_{\text{эфф}}}{\nu} = \frac{G_{\rho} \cdot \delta_{\text{эфф}}}{\rho S_{\text{кан}} \cdot \nu},$$

$$v_{\text{кан}} = \frac{G_{\rho}}{\rho S_{\text{кан}}},$$

где  $v_{\text{кан}}$ , м/с – скорость воздуха в каналах ЭВМ, образованных ЭМ и ИЭП;  $\lambda$ , Вт/(м К),  $\rho$ , кг/м<sup>3</sup>,  $\nu$ , м<sup>2</sup>/с – теплопроводность, плотность и кинематическая вязкость воздуха ( $\lambda = 0,0283$  Вт/(м·К),  $\rho = 1,93$  кг/м<sup>3</sup>,  $\nu = 17,95 \cdot 10^{-6}$  при +50°C);  $S_{\text{кан}}$  – суммарная площадь поперечного сечения каналов ЭВМ (19 каналов, имеющих эффективную ширину 9,1 мм (см. выше) и глубину  $L_{1,ЭМ} = 150$  мм, равную ширине ЭМ и ИЭП (перпендикулярной направлению потока воздуха), участвующей в теплообмене с воздухом,  $S_{\text{кан}} = 19 \cdot \delta_{\text{эфф}} \cdot L_{1,ЭМ} = 19 \cdot 9,1 \cdot 150 = 25935$  мм<sup>2</sup>).

Для вычисления критерия Нуссельта  $Nu$  оценивается длина начального аэродинамического участка  $l_n$  и сопоставляется с длиной канала по направлению движения потока воздуха  $L_{2,ЭМ} = 233$  мм [3 – 7], а именно,  $l_n = 0,01 \cdot \delta_{\text{эфф}} \cdot Re$ . Если  $l_n < L_{2,ЭМ}$ , то критерий Нуссельта вычисляется как  $Nu = 4,12 \cdot (1 + 0,5 \cdot l_n / L_{2,ЭМ})$ .

Результаты расчетов средней температуры воздуха  $T_v$  в каналах ЭВМ и температуры воздуха  $T_{v,вых}$  на выходе из каналов, приведены на рис. 5. Так, для того чтобы средняя температура воздуха в каналах ЭВМ составляла 60°C, а температура воздуха на выходе из каналов составляла 70°C, массовый расход воздуха должен быть равным 225 кг/ч.

#### 4.2 Результаты расчета параметров аэродинамического и теплового режимов корпуса микропроцессора и корпуса ИЭП

Параметры аэродинамического и теплового режимов МС микропроцессора, с установленным на его корпусе штыревым радиатором, размеры которого приведены в п. 3.1, рассчитываются по следующим формулам:

– тепловое сопротивление радиатора  $R_{\text{рад,шт}}$  равно [4]

$$R_{\text{рад,шт}} = cth(bh') / (\lambda_{\text{рад}} S_{1,шт} b N),$$

$$b = \sqrt{\alpha_{\text{рад}} p_{1,шт} / (\lambda_{\text{рад}} S_{1,шт})}, \quad h' = h + S_{1,шт} / p_{1,шт},$$

где  $\alpha_{\text{рад}}$  – коэффициент теплоотдачи  $\alpha_{\text{рад}}$  вычисляется по формуле [4, 7]:

$$\alpha_{\text{рад}} = Nu \frac{\lambda_{\text{возд}}}{a}.$$

Полная тепловая проводимость ( $\sigma_{\text{рад}}$ ) штыревого радиатора складывается из

тепловых проводимостей штыревой поверхности радиатора  $\sigma_{\text{рад,шт}} = 1/R_{\text{рад,шт}}$  и неоребреной поверхности  $\sigma_{\text{рад,необр}} = \alpha_{\text{рад}} S_{\text{осн,рад}}$ , то есть

$$\sigma_{\text{рад}} = \sigma_{\text{рад,шт}} + \sigma_{\text{рад,необр}},$$

где  $S_{\text{осн,рад}} = l^2 = 58^2 = 3364$  мм<sup>2</sup> – площадь основания радиатора.

Критерий Нуссельта для штыревого радиатора  $\alpha_{\text{рад}}$  на корпусе процессора определяется [5, 7]:



Рис. 6. Результаты расчета зависимостей температуры  $T_{p-c}$  на радиаторе корпуса микропроцессора и температуры  $T_{p-корп.ИЭП}$  на радиаторах корпуса ИЭП от массового расхода воздуха через корпус

- при  $Re = \frac{v_{\text{кан}} \cdot \delta_{\text{эфф}}}{\nu} < 1000$ , по формуле  $Nu_{\text{рад}} = 0,49 Re^{0,5}$ ,
- при  $Re = \frac{v_{\text{кан}} \cdot \delta_{\text{эфф}}}{\nu} > 1000$ , по формуле  $Nu_{\text{рад}} = 0,194 Re^{0,65}$ .

Аналогично вычисляются и параметры аэродинамического и теплового режимов конструкции ИЭП, боковые пластины которых выполнены в виде штыревых радиаторов.

Результаты расчета температуры  $T_{p-c}$  на радиаторе, прижимаемом к корпусу процессора, и на радиаторах корпуса ИЭП  $T_{p-корп.ИЭП}$ , приведены на рис. 6. Так, при массовом расходе воздуха равном 225 кг/ч и температуре воздуха на входе в ЭВМ равной +50°C, температура на радиаторе корпуса процессора составит 65°C, а температура на радиаторах корпуса конструкции ИЭП будет равна 71°C.

Отметим, что указанные значения температуры на радиаторе, прижимаемом к корпусу микропроцессора, получены при условии идеального теплового контакта

между контактирующими поверхностями. В реальности контактное тепловое сопротивление присутствует всегда. Для создания минимальных значений температуры на корпусе микропроцессора и на его кристалле, необходимо в место контакта вводить теплопроводную прокладку (например, фольгу ТРГ) или теплопроводную пасту (например, Arctic Silver 5). Однако введение прокладок и паст между контактирующими поверхностями требует создания значительных усилий прижатия, что не всегда является приемлемым в конкретной конструкции модуля. Между тем, поскольку конструкция ЭВМ не подвергается воздействию дестабилизирующих факторов (вибрации, ускорения и пр.), то можно рекомендовать клеевое соединение радиатора с корпусом МС микропроцессора с помощью высокотеплопроводного клея.

## 5. Заключение

Для эффективного охлаждения высокопроизводительных защищенных серверов с воздушным охлаждением, имеющих архитектуру как у рассматриваемой ЭВМ, необходимо осуществлять техническое проектирование их конструкции в следующем направлении:

1) Установить на корпусе МС микропроцессора штыревой радиатор. Спроектированный штыревой радиатор должен прижиматься к корпусу (крышке корпуса) процессора (см. рис. 2), при этом контакт между корпусом и радиатором осуществляется через теплопроводную прокладку (например, фольгу ТРГ) или теплопроводную пасту (например, Arctic Silver 5) при создании соответствующего прижима.

Штыревой радиатор должен иметь следующие параметры:

- материал штыревого радиатора медь с коэффициентом теплопроводности ( $\lambda_{\text{рад}} = 395$  Вт/мК);
- основание радиатора – квадрат со стороной  $l = 58$  мм;
- толщина основания радиатора  $d = 3$  мм;
- сечение штыря – квадрат, со стороной штыря  $\delta_{\text{шт}} = a = 2$  мм;
- высота штыря  $h = 8$  мм;
- расстояние между штырями  $b = 2$  мм;
- площадь сечения штыря  $S_{1,\text{шт}} = a^2 = 4$  мм<sup>2</sup>;
- периметр сечения штыря  $p_{1,\text{шт}} = 8$  мм;
- количество штырей радиатора  $N$ , размещаемых на прямоугольном основании радиатора с размерами  $l = 58$  мм равно  $N = 225$  шт.

2) Выполнить с двух сторон корпуса ИЭП штыревые радиаторы в виде единой конструкции с боковыми пластинами корпуса ИЭП (см. рис. 3).

Штыревой радиатор с каждой стороны корпуса ИЭП должен иметь следующие параметры:

- материал штыревого радиатора (а также боковых пластин ИЭП) – алюминиевый сплав с коэффициентом теплопроводности ( $\lambda_{\text{рад}}$ ) не менее 130 Вт/(м·К) (далее в расчетах радиатора принято  $\lambda_{\text{рад}} = 130$  Вт/(м·К));
- штыри радиатора выполнены в единой конструкции с боковыми пластинами корпуса ИЭП с вертикальными и горизонтальными размерами (по направлению движения потока воздуха) пластин 223 мм и 150 мм;
- толщина основания радиатора – боковой пластины корпуса ИЭП, равно  $d = 2$  мм;
- сечение штыря – квадрат, со стороной штыря  $\delta_{\text{шт}} = a = 3$  мм;
- высота штыря  $h = 4$  мм;
- расстояние между штырями  $b = 5$  мм;
- площадь сечения штыря  $S_{1,\text{шт}} = a^2 = 9$  мм<sup>2</sup>;
- периметр сечения штыря  $p_{1,\text{шт}} = 12$  мм;
- количество штырей радиатора  $N$ , размещаемых на прямоугольном основании радиатора 223 мм × 150 мм ( $\delta_{\text{шт}} = a$ ) равно  $N = 551$  шт. При этом число штырей по сторонам 223 мм и 150 мм укладывается целое число штырей 29 и 19, соответственно.

*Примечание:* алюминиевый сплав АМГ6 имеет низкую теплопроводность (90 Вт/(м·К)) и его применение приведет к снижению теплоотводящих характеристик штыревых радиаторов корпуса ИЭП.

3) Для обеспечения эффективной теплопередачи от корпуса процессора к штыревому радиатору необходимо обеспечить низкую величину контактного теплового сопротивления. С этой целью необходимо:

- обеспечить толщину контактных зазоров не более 0,2 мм при высокой чистоте обработки контактной поверхности радиатора, малых значениях неплоскостности и волнистости поверхности радиатора;
- заполнять зазоры между контактирующими поверхностями корпуса и радиатора высокотеплопроводными прокладками (например, фольга из терморасширенного графита (ТРГ)) или высокотеплопроводными пастами (например, Arctic Silver 5);
- обеспечить значительные усилия болтового прижатия всех контактирующих поверхностей.

4) Выполнение приведенных рекомендаций по конструкции, изготовлению и монтажу штыревых радиаторов в ЭМ и ИЭП в ЭВМ обеспечит следующие параметры теплового и аэродинамического режимов

ЭВМ при температуре среды на входе в ЭВМ +50 °С (при идеальном тепловом контакте между контактирующими поверхностями):

- средняя температура воздуха  $T_v$  в каналах, образованных ЭМ и ИЭП для массового расхода воздуха  $G_p \geq 225$  кг/ч, будет меньше  $T_v \leq 60^\circ\text{C}$ ;

- температура воздуха на выходе из каналов  $T_{v,\text{вых}}$  для массового расхода воздуха  $G_p \geq 225$  кг/ч, будет меньше  $T_{v,\text{вых}} \leq 70^\circ\text{C}$  (см. рис. 5);

- температура  $T_{p-c}$  на радиаторе, установленном на корпусе (крышке корпуса) микропроцессора (при идеальном контакте между корпусом и радиатором) для массового расхода воздуха  $G_p \geq 225$  кг/ч, будет меньше  $T_c \leq 65^\circ\text{C}$  (см. рис. 6);

- температура  $T_{p-\text{корп.ИЭП}}$  на радиаторах корпуса ИЭП для массового расхода воздуха  $G_p \geq 225$  кг/ч, будет меньше  $T_{\text{корп.ИЭП}} \leq 71^\circ\text{C}$  (см. рис. 6);

- поскольку конструкция ЭВМ не подвергается воздействию серьезных дестабилизирующих факторов (вибрации, ускорения и пр.), то можно рекомендовать

выполнить контакт между радиатором и крышкой корпуса процессора в виде клевого соединения с помощью клея с высокой теплопроводностью.

5) Для уменьшения проходного сечения воздуха при входе в ЭВМ и соответствующего увеличения скорости потока воздуха через каналы, образованными ЭМ и ИЭП, рекомендуется, ввести в конструкцию ЭМ вентиляционную решетку (с 16-ю просветами шириной 9 мм и длиной 150 мм, равной ширине ЭМ и ИЭП, перпендикулярной направлению потока воздуха), перекрывающую своими глухими частями свободное пространство (просвет) между соседними ЭМ (см. рис. 4).

Публикация выполнена в рамках выполнения госзадания НИР «Разработка общих принципов и методик создания корпусов моноблоков и серверов в соответствии с требованиями по уровням электромагнитных излучений и отводу тепла» (0065-2018-0018).

## Methods for ensuring heat dissipation in enclosures of high-performance secure servers with air cooling

A.V. Bakaldin, P.I. Kandalov, N.A. Kozlov, A.G. Madera, O.V. Serdin

**Abstract:** The transfer, accumulation, analysis and distribution of big volume of data place high demands on the speed of their processing, the bandwidth of signals, stability and noise immunity. The high-performance computer systems designed to meet these challenges, based on VPX (VITA 46) and penVPX (VITA 65) specifications, along with high bandwidth and noise immunity, have an increased power consumption. This leads to significant heat release, which are an obstacle to the creation of high-efficient and competitive systems. In this paper, methods for ensuring thermal modes for the high-performance, secure servers with forced air cooling are considered. Specific designs of heat sink systems, methods of thermal design and the results obtained are presented.

**Keywords:** big data, high-performance computing systems, thermal modes, protected server, heat sink, microprocessor, air cooling

### Литература

1. А.Г. Мадера, П.И. Кандалов. Моделирование трехмерных температурных полей в электронных модулях // Программные продукты и системы. 2010. №2. С. 36
2. А.Г. Мадера Концепция математического и компьютерного моделирования тепловых процессов в электронных системах // Программные продукты и системы. 2015. № 3.С. 79 – 86
3. В.М. Белецкий, Г.А. Кривов. Алюминиевые сплавы (состав, свойства, технология, применение). Справочник / Под общей ред. ак. РАН И.Н. Фридляндера. – Киев: Коминтех, 2005.
4. Г.Н. Дульнев Тепло- и массообмен в радиоэлектронной аппаратуре. – М. Высш. Шк., 1984.
5. Г.Н. Дульнев, Э.М. Семяшкин. Теплообмен в радиоэлектронных аппаратах. – Л.: Энергия, 1968.
6. А.Г. Мадера Моделирование теплообмена в технических системах. – М.: Науч. фонд «Первая исслед. лаб. им. акад. В.А. Мельникова», 2005
7. М.А. Михеев, И.М. Михеева. Основы теплопередачи. – М.: Бастед, 2010.

# Современные технологии охлаждения с применением контурных тепловых труб в корпусах бортовых серверов с повышенной мощностью потребления

А.В. Бакалдин<sup>1</sup>, А.М. Баранов<sup>2</sup>, П.И. Кандалов<sup>3</sup>, А.Г. Мадера<sup>4</sup>, О.В. Сердин<sup>5</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

E-mail's: <sup>1</sup>[bakaldin@cs.niisi.ras.ru](mailto:bakaldin@cs.niisi.ras.ru), <sup>2</sup>[baranov@niisi.ras.ru](mailto:baranov@niisi.ras.ru),

<sup>3</sup>[petrki87@gmail.com](mailto:petrki87@gmail.com), <sup>4</sup>[omm@niisi.ras.ru](mailto:omm@niisi.ras.ru), <sup>5</sup>[serdin@cs.niisi.ras.ru](mailto:serdin@cs.niisi.ras.ru)

**Аннотация:** Бортовые серверы, обрабатывающие большие потоки данных, функционируют в жестких окружающих условиях, воздействию дестабилизирующих факторов и значительным энергопотреблением. Для обеспечения заданного теплового режима сервера необходимо применять современные технологии охлаждения, использующие контурные тепловые трубы, способные отводить мощность потребляемую микропроцессором до 200 Вт. В данной работе проанализированы методики создания корпусов серверов с использованием современных технологий охлаждения с применением контурных тепловых труб. Рассматривается тепловой режим конструкции на примере бортового сервера с десятью электронными модулями и источником питания общей мощностью 800 Вт, размещенных в корпусе, герметичном к проникновению электромагнитных излучений, влаги, грязи и пыли.

**Ключевые слова:** бортовой сервер, ЭВМ, охлаждение, контурная тепловая труба, тепловые режимы, отвод тепла, микропроцессор, корпус ЭВМ, пониженное давление

## 1. Введение

Бортовые серверы, являющиеся основой современных высокопроизводительных систем обработки и анализа больших потоков данных, изображений и информации в режиме реального времени, эксплуатируются в жестких окружающих условиях и воздействию многочисленных дестабилизирующих факторов.

С ужесточением требований, предъявляемым к эксплуатационным параметрам серверов, увеличением их производительности и мощности потребления [1], растут также требования к системам обеспечения теплового режима, охлаждению и отвода тепла.

Однако в силу значительного тепловыделения современных конструкций серверных систем, традиционные способы охлаждения с использованием естественной и принудительной конвекции, радиаторов, металлических теплоотводов, не позволяют существенно снизить уровни температуры в системе. Поэтому в настоящее время возникает все большая потребность в современных технологиях охлаждения с применением тепловых труб, и, в частности, контурных тепловых труб (КТТ), на практике показавших свое преимущество в сравнении с другими способами охлаждения и отвода тепла, позволяющих обеспечивать вы-

сокоэффективный отвод тепла от микропроцессоров с высоким тепловыделением [2, 3, 4].

Принятые для разработки конструкции бортовых серверов основаны на спецификациях VPX (VITA 46) и OpenVPX (VITA 65) и их отличительными особенностями являются: модульность, повышающая ремонтпригодность; высокая пропускная способность объединительной панели; отсутствие чувствительности к электромагнитным помехам; высокое энергопотребление входящих в состав электронных модулей (максимальное потребление модуля при напряжении питания 5 В постоянного тока составляет 115 Вт, при 12 В – 384 Вт); надежное функционирование в жестких и экстремальных условиях эксплуатации.

Потребляемая мощность высокопроизводительных бортовых серверов может достигать 2 КВт и более, при этом сервер должен функционировать под действием значительных механических нагрузок, температурах окружающей среды от  $-60$  °С до  $+55$  °С, пониженном атмосферном давлении. Для обеспечения требуемого теплового режима бортового сервера необходимо использование современных способов охлаждения на основе КТТ.

В работе проанализированы общие принципы, методики и способы обеспечения отвода тепла с использованием КТТ в корпу-

сах высокопроизводительных бортовых серверов. В качестве примера рассматривается бортовой сервер, в состав которого входят десять электронных модулей и источник питания общей мощностью 800 Вт, размещенные в корпусе, герметичном к проникновению электромагнитных излучений, влаги, грязи и пыли. Тепловое проектирование корпусов сервера проводилось по методикам [5, 6, 7].

## 2. Конструкция и принципы охлаждения сервера

Рассматриваемый сервер (далее ЭВМ)

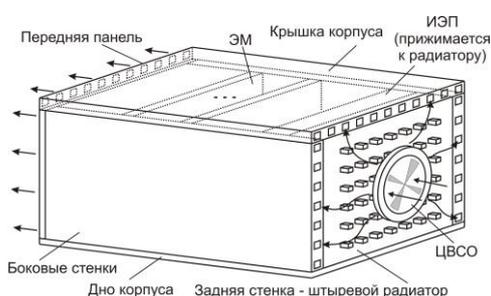


Рис. 1. Конструкция корпуса ЭВМ с каналами в боковых стенках и крышке корпуса и выполненным на задней стенке корпуса радиатором, к основанию которого прижимается модуль ИЭП.

конструктивно выполнен в герметичном корпусе, который по своим тепловым и аэродинамическим характеристикам должен обеспечивать эффективное охлаждение электронных модулей (ЭМ) и источников электропитания (ИЭП), размещенных внутри него, с учетом требований по тепловому режиму и при воздействии дестабилизирующих климатических и тепловых факторов (табл. 1).

Таблица 1. Воздействующие климатические и тепловые факторы

Наименование воздействующих факторов	Значение параметра
Повышенная относительная влажность воздуха при $T=35^{\circ}\text{C}$ , %	100
Повышенная температура среды, $^{\circ}\text{C}$	+55
Пониженная температура среды, $^{\circ}\text{C}$	- 60
Пониженное атмосферное давление, Па (мм.рт.ст.)	2000 (15)

Потребляемая ЭВМ мощность (табл. 2) составляет  $P_{\text{ЭВМ}} = 800$  Вт и корпус ЭВМ должен обеспечить эффективное охлаждение размещенных внутри него ЭМ и ИЭП как при

заданной мощности, так и воздействию дестабилизирующих факторов (табл. 1).

Конструкция корпуса ЭВМ (рис. 1) содержит боковые стенки, крышку, заднюю стенку, дно и переднюю панель с электрическими соединителями. На дне корпуса со стороны внутреннего объема устанавливается объединительная панель. Конструкции ЭМ и ИЭП устанавливаются в массивных металлических пазах-направляющих, выполненных в боковых стенках корпуса, и прижимающихся к ним с помощью клинового механизма.

Отвод тепла от ЭМ и ИЭП на корпус ЭВМ осуществляется путем кондукции через контактные площадки между ЭМ и направляющими, обеспечивая тем самым кондуктивный отвод тепла. Для охлаждения корпуса ЭВМ в боковых стенках и крышке корпуса выполнены каналы, по которым нагнетается воздушный поток от централизованной воздушной системы охлаждения (ЦВСО).

Таблица 2. Потребляемые мощности

Наименование	Количество, шт	P, Вт
ЭМ с двумя микропроцессорами*	10	60
ИЭП	1	200
Суммарная мощность ЭВМ		800

\* В рассматриваемой ЭВМ используются микропроцессоры с архитектурой КОМДИВ64 с мощностью потребления 25 Вт.

Предварительный анализ теплового и аэродинамического режимов показал, что для эффективного охлаждения ЭВМ необходимо провести тепловое техническое проектирование конструкции корпуса в следующих направлениях:

1) Оптимизировать размеры каналов теплообменника корпуса с точки зрения обеспечения как достаточной для теплообмена (при соответствующих ограничениях) площади поверхности, так и низкого значения аэродинамического сопротивления каналов нагнетаемому воздуху.

2) Для эффективного кондуктивного растекания тепла выполнить толщину боковых стенок, крышки и задней стенки корпуса равной не менее 12 мм.

3) Ввиду значительной величины мощности потребления ИЭП (200 Вт), необходимо обеспечить надежный тепловой контакт модуля ИЭП с задней стенкой корпуса, причем заднюю стенку со стороны нагнетания воздуха, выполнить в виде развитого штыревого радиатора (рис. 1).

4) В конструкции корпуса ЭВМ необходимо применять сплав алюминия с высоким значением теплопроводности. Возможные для

применения в конструкции корпуса алюминиевые сплавы приведены в [8]. Материал корпуса должен выбираться исходя как из высокой теплопроводности, так и из конструктивных свойств с точки зрения удовлетворения воздействующим факторам (прочность, твердость, и др.).

*Примечание:* алюминиевый сплав АМг6

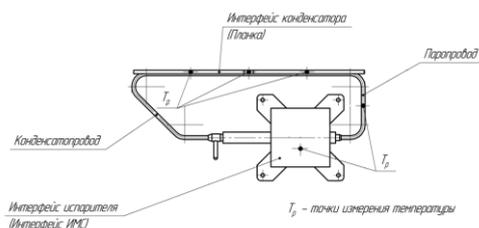


Рис. 2. Контурная тепловая труба, устанавливаемая на корпусе микропроцессора (разработка ООО «Теркон-КТТ»)

имеет низкую теплопроводность (90 Вт/(м·К)) и его применение не рекомендуется.

5) В ЭМ, входящих в состав ЭВМ, установлено по две микросхемы (МС) микропроцессора с мощностью потребления 25 Вт (табл. 2); остальные МС и электрорадиоэлементы (ЭРЭ), потребляют в сумме ~10 Вт. Ввиду значительной мощности тепловыделения как микропроцессора, так и ЭМ как уже указывалось выше, необходимо применение современных методов теплоотвода на основе КТТ [2, 3, 4]. Конструкции КТТ представляют собой высокоэффективные теплопередающие устройства, позволяющие отводить мощности до 2 кВт, на расстояние до 20 м, при произвольной ориентации в пространстве, нулевом уровне шума и сроком службы до 20 лет.

Таблица 3. Технические характеристики КТТ

Характеристика	Значение
Максимальная мощность, подаваемая на интерфейс, Вт	35
Теплоноситель	Аммиак
Размеры источника тепла, мм×мм	31 × 31
Размеры интерфейса конденсатора (планки), мм×мм	125 × 12
Материал интерфейса испарителя	Медь
Материал интерфейса конденсатора	Медь
Тепловое сопротивление, К/Вт	< 0,65
Время запуска, с	< 10

Для передачи тепла от микропроцессоров на корпус ЭВМ могут быть использованы КТТ, разработанные ООО «Теркон-КТТ» (рис. 2), основные технические характеристики которых приведены в табл. 3.

Мощность, потребляемая микропроцессором, преобразуется в теплоту, которая рассеивается по корпусу микропроцессора и поступает от его кристалла, через теплопроводный компаунд между кристаллом и крышкой корпуса, – на крышку корпуса и далее растекается по его конструкции. Тепловой поток от крышки корпуса микропроцессора передается через высокотеплопроводный слой к прижатому к крышке интерфейсу испарителя, входящему в состав конструкции КТТ, и далее передается через устройство КТТ к интерфейсу конденсатора (рис. 2 и рис. 3), припаянного к теплопроводной планке, которая, в свою очередь, прижимается к рамке ЭМ. Тепловые потоки от рамки ЭМ передаются к направля-

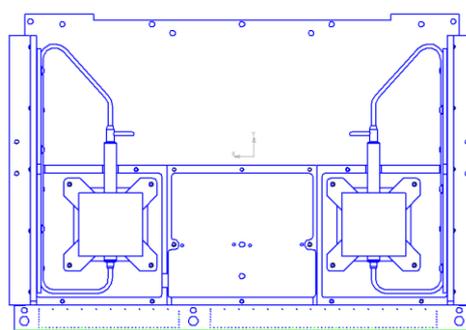


Рис. 3. ЭМ с двумя микропроцессорами, в которых отвод тепла осуществляется посредством КТТ

ющим пазам корпуса ЭВМ, кондуктивно растекаются по корпусу и далее уносятся воздухом, нагнетаемым от ЦВСО через каналы в боковых стенках и крышке корпуса (рис. 1).

Кроме двух микропроцессоров общей мощностью потребления 50 Вт (рис. 3), на плате ЭМ установлены МС и ЭРЭ, с суммарной мощностью тепловыделения 10 Вт. Тепловые потоки от МС и ЭРЭ растекаются по многослойной печатной плате (МПП) и далее через направляющие пазы передаются на каркас корпуса ЭВМ и уносятся, нагнетаемым через каналы, воздухом. Процессы естественной конвекции внутри герметичного корпуса ЭВМ практически не развиваются и ими в расчетах можно пренебречь.

### 3. Предельно допустимые температуры на корпусе микропроцессора и на внутренних поверхностях корпуса ЭВМ

Большая часть потребляемой в ЭМ мощности, приходится на микропроцессоры, поэтому температура на кристалле и корпусе

микропроцессора является критической и определяет предельные уровни температуры конструкции корпуса ЭВМ (рис. 1). Определим предельно допустимую температуру  $T_{c,max}$  на корпусе микропроцессора и предельно допустимую температуру  $T_{ЭВМ,max}$  на внутренней поверхности корпуса ЭВМ (боковых стенках и крышке корпуса, пазах-направляющих).

### 3.1 Предельно допустимая температура на корпусе микропроцессора

В рассматриваемой ЭВМ используются микропроцессоры, предельная температура р-р-перехода кристалла которых равна  $T_{j,max} = 150^\circ\text{C}$ . Примем с запасом, что предельно допустимая температура кристалла микропроцессора составляет  $T_{j,max} = 125^\circ\text{C}$ . Будем также полагать, что тепловое сопротивление корпуса  $R_{jc}$ , не превышает величины  $0,7 \text{ К/Вт}$ .

Приняв, что вся мощность тепловыделения микропроцессора ( $P_{МС} = 25 \text{ Вт}$ ) поступает к крышке корпуса и далее отводится от нее посредством КТТ, получим предельно допустимую температуру на корпусе МС:

$$T_{c,max} \leq T_{j,max} - P_{МС} \cdot R_{jc} = 107,5^\circ\text{C}.$$

Примем значение предельно допустимой температуры на корпусе микропроцессора равным  $105^\circ\text{C}$ , то есть

$$T_{c,max} = 105^\circ\text{C}.$$

### 3.2 Предельно допустимая температура на внутренних поверхностях корпуса ЭВМ

Рассчитаем предельно допустимую температуру  $T_{ЭВМ,max}$  на внутренней поверхности корпуса ЭВМ, приняв, что вся мощность тепловыделения от микропроцессора ( $P_{МС} = 25 \text{ Вт}$ ) поступает к крышке его корпуса и отводится далее посредством КТТ к внутренним стенкам корпуса ЭВМ (пазах-направляющим).

Предельно допустимая температура ( $T_{ЭВМ,max}$ ) корпуса ЭВМ определяется согласно выражению:

$$T_{ЭВМ,max} \leq T_{c,max} - P_{МС} \cdot R_{тракта}.$$

Здесь  $R_{тракта}$  – полное тепловое сопротивление тракта, по которому отводится тепловой поток от корпуса микропроцессора, которое складывается из следующих тепловых сопротивлений:

– теплового контактного сопротивления ( $R_{МС-ИнтИМС}$ ) между поверхностью корпуса МС микропроцессора и интерфейсом ИМС (рис. 2);

– теплового сопротивления ( $R_{ИнтИМС}$ ) интерфейса ИМС (рис. 2), выполненного из меди;

– теплового сопротивления КТТ  $R_{КТТ}$  (табл. 3);  
– теплового сопротивления ( $R_{Планка}$ ) планки (интерфейс конденсатора), выполненной из меди;

– теплового контактного сопротивления ( $R_{Планка-ЭВМ}$ ) между планкой (рис. 2) и рамкой ЭМ (рис. 3), контакт которого заполнен теплопроводным компаундом.

Тепловые сопротивления равны:

$$R_{МС-ИнтИМС} = \frac{\delta_{МС-ИнтИМС}}{\lambda_{ТРГ} \cdot S_{c-r}} = 0,0278 \text{ К/Вт};$$

$$R_{ИнтИМС} = \frac{\delta_{ИнтИМС}}{\lambda_{медь} \cdot S_{c-r}} = 0,0225 \text{ К/Вт};$$

$$R_{Планка} = \frac{\delta_{Планка}}{\lambda_{медь} \cdot S_{Планка}} = 0,00534 \frac{\text{К}}{\text{Вт}};$$

$$R_{Планка-ЭВМ} = \frac{\delta_{Планка-ЭВМ}}{\lambda_{ТРГ} \cdot S_{Планка}} = 0,01758 \text{ К/Вт}.$$

Тепловое сопротивление тракта составит:

$$R_{тракта} = 0,7232 \text{ К/Вт}.$$

Перепад температуры  $\Delta T_{c-ЭВМ}$  ( $\Delta T_{c-ЭВМ} = T_c - T_{ЭВМ}$ ), равный разности температуры  $T_c$  корпуса микропроцессора (крышкой корпуса) и температуры  $T_{ЭВМ}$  корпуса ЭВМ (его внутренней поверхности) составляет величину:

$$\Delta T_{c-ЭВМ} = P_{МС} \cdot R_{тракта} = 18^\circ\text{C}.$$

Тогда предельно допустимая температура корпуса ЭВМ ( $T_{ЭВМ,max}$ ), при которой температура на корпусе микропроцессора ( $T_{c,max}$ ) не превысит  $105^\circ\text{C}$ , составит:

$$T_{ЭВМ,max} = T_{c,max} - P_{МС} \cdot R_{тракта} = 87^\circ\text{C}.$$

Примем значение предельно допустимой температуры корпуса ЭВМ равным  $T_{ЭВМ,max} = 85^\circ\text{C}$ .

## 4. Результаты расчетов теплового и аэродинамического режимов конструкции корпуса ЭВМ при нормальном и пониженном давлении

Тепловые потоки от ЭМ рассеиваются по боковым стенкам и крышке корпуса ЭВМ, поступают к нагнетаемому по каналам в стенках и крышке воздуху, нагревают его и уносятся потоком воздуха за пределы корпуса (рис. 1). Особенность процессов теплопереноса в конструкции корпуса ЭВМ состоит в том, что теплопередача от МС и ЭРЭ, установленных на МПП в ЭМ, а также теплопередача от ЭМ к стенкам и крышке корпуса ЭВМ, осуществляются преимущественно посредством теплопроводности (кондукции). При этом потоки тепла от стенок и крышки корпуса передаются к каналам в боковых стенках и крышке корпуса посредством вынужденной конвекции

потоку воздуха, нагнетаемого через каналы. Так как корпус ЭВМ является герметичным, процессы естественной конвекции внутри объема корпуса практически не развиваются, и ими при моделировании можно пренебречь.

Кондуктивный характер теплопередачи в конструкции корпуса ЭВМ накладывает жесткие требования к величине контактных тепловых сопротивлений вдоль всего тракта протекания потоков тепла к продуваемому через каналы воздуху. Это обуславливается тем, что существенный вклад в тепловые потери, приводящие к недопустимым перегревам элементов ЭВМ (МС, ЭРЭ, ЭМ) и всей конструкции ЭВМ, вносится именно контактными тепловыми сопротивлениями. Поэтому все контактные тепловые сопротивления между контактирующими частями конструкции должны иметь минимально возможные значения. К критически важным для уровней температуры МС, ЭРЭ и корпуса ЭВМ контактными тепловыми сопротивлениям относятся контакты между следующими элементами конструкции:



Рис. 4. Боковые стенки, крышка корпуса с каналами, через которые нагнетается воздух от ЦВСО и задняя стенка корпуса ЭВМ с штыревым радиатором

Обозначения: вк, шк – высота и ширина каналов; т-кр, т-ст – толщина растекателя тепла крышки и стенки; п – ширина перегородки между каналами; Шз.ст, Вз.ст – ширина и высота задней стенки

- МС и интерфейсом испарителя КТТ (рис. 2);
- интерфейсом конденсатора КТТ (планкой) и рамкой ЭМ (рис. 3);
- рамкой модуля и направляющими пазами;
- крышкой корпуса ЭВМ и боковыми стенками (рис. 1);
- крышкой корпуса ЭВМ, боковыми стенками и задней стенкой корпуса (рис. 1).

Уменьшение контактных тепловых сопротивлений на практике достигается, во-первых, проектированием конструкции с малыми толщинами зазоров (не более 0,2 мм) и

высокими требованиями к чистоте обработки поверхностей, величинам неплоскостности и волнистости, во-вторых, введением в зазоры между контактирующими поверхностями высокотеплопроводных прокладок (например, эластосил, фольга из ТРГ) или высокотеплопроводных паст (например, Arctic Silver 5), в-третьих, значительными усилиями болтового соединения контактирующих поверхностей.

Для создания конструкции, обеспечивающей эффективное охлаждение ЭВМ было проведено тепловое техническое проектирование корпуса на основании расчетов тепло-

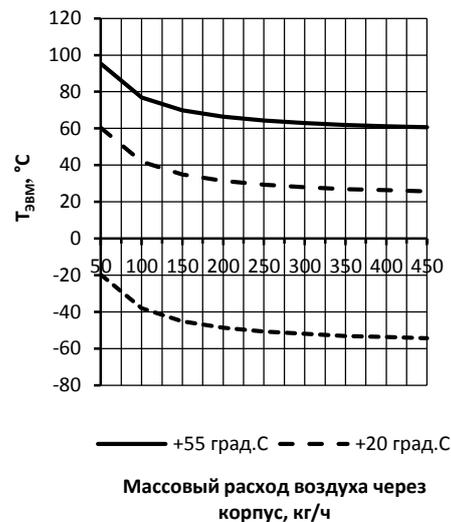


Рис. 5. Рассчитанные температуры ( $T_{ЭВМ}$ ) корпуса ЭВМ в зависимости от массового расхода воздуха через корпус при температурах среды +55°C, +20°C, -60°C и нормальном давлении

вых и аэродинамических режимов в рабочем диапазоне температур окружающей воздушной среды от -60°C до +55°C как при нормальном, так и пониженном давлении 15 мм.рт.ст. При этом диапазон температуры среды на входе в каналы корпуса ( $T_{Возд.Вх}$ ) составляет от -60°C до +55°C и является рабочим температурным диапазоном окружающего воздуха, нагнетаемого в каналы корпуса.

Из-за значительной величины потребляемой мощности модуль ИЭП устанавливается в пазах корпуса, примыкающим к задней стенке корпуса, и прижимается к ней изнутри винтами через высокотеплопроводную прокладку или высокотеплопроводную пасту. При этом внешняя сторона задней стенки корпуса выполнена в виде штыревого радиатора, который охлаждается потоком воздуха (рис. 4).

Результаты расчета температуры ( $T_{ЭВМ}$ ) корпуса ЭВМ в зависимости от массового расхода воздуха через корпус при температу-

рах среды +55 °С, +20 °С, -60 °С и нормальном давлении, для принятых параметрах конструкции корпуса ЭВМ и нормальном давлении, приведены на рис. 5.

Согласно техническим требованиям (см. табл. 1) электронные модули должны сохранять работоспособность при пониженных атмосферных давлениях окружающей среды 2000 Па (15 мм.рт.ст.).

При давлениях, отличных от нормального, плотность воздуха, поступающего для охлаждения ЭВМ ( $T_{\text{Возд.вх}}$ ), вычисляется по формуле [9, 10]:

$$\rho = \frac{353,2 \cdot p}{T_{\text{Возд.вх}} \cdot p_0},$$

где  $p_0$  – давление воздуха при нормальных условиях,  $p_0 = 101$  кПа (760 мм.рт.ст.).

Результаты расчета температуры корпуса ЭВМ при пониженном давлении 2000 Па



Рис. 6. Рассчитанные температуры ( $T_{\text{ЭВМ}}$ ) корпуса ЭВМ в зависимости от массового расхода воздуха при температурах среды +55 °С, +20 °С, -60 °С и пониженном давлении 2000 Па (15 мм.рт.ст.).

(15 мм.рт.ст.) и температурах среды +55 °С, +20 °С, -60 °С приведены на рис. 6.

Из полученных зависимостей следует, что для того чтобы температура на корпусе ЭВМ не превышала предельно допустимую ( $T_{\text{ЭВМ,max}} = 85^\circ\text{C}$ ) при пониженном давлении 2000 Па (15 мм.рт.ст.) и температуре среды +55 °С, необходимо обеспечить расход воздуха через корпус не менее  $G_p \geq 120$  кг/ч и перепад давления между входом и выходом корпуса не меньше, чем  $\Delta p \geq 4500$  Па при пониженном давлении 2000 Па (15 мм.рт.ст.) и температуре среды +55 °С.

Эффективность охлаждения корпуса ЭВМ при пониженном давлении 2000 Па (15 мм.рт.ст.) может быть повышена, если рас-

смотренную выше конструкцию корпуса (с конвективно-кондуктивным воздушным способом охлаждения) изменить на конструкцию, в которой реализуется исключительно кондуктивный принцип отвода тепла по массивным стенкам корпуса, прижимаемым к охлаждаемым панелям, которые поддерживаются при заданной температуре.

## 5. Заключение

Для эффективного охлаждения высокопроизводительных бортовых серверов с рассматриваемой архитектурой, необходимо при техническом тепловом проектировании конструкции следовать следующим рекомендациям:

1) Развить площадь воздушного теплообменника корпуса ЭВМ, состоящего из боковых стенок и крышки корпуса, выполнив каналы для нагнетания воздуха как в боковых стенках, так и в крышке.

2) ИЭП устанавливать на мощном основании-растекателе тепла из материала с высокой теплопроводностью. Необходимо обеспечить также значительное прижатие (через высокотеплопроводящие прокладки или пасты) основания-растекателя модуля ИЭП к внутренней поверхности задней стенки корпуса.

3) Заднюю стенку корпуса со стороны поступления охлаждающего воздуха выполнить в виде штыревого радиатора, имеющего следующие параметры:

– материал штыревого радиатора (задней стенки корпуса) – алюминиевый сплав с коэффициентом теплопроводности ( $\lambda_{\text{рад}}$ ) не менее 130 Вт/(м·К);

– количество штырей радиатора, размещаемых на прямоугольном основании радиатора, выполненного на задней стенке корпуса ЭВМ составляет не менее 1650 шт.

4) Корпус ЭВМ необходимо выполнить из высокотеплопроводных сплавов алюминия, теплопроводностью не менее 130 Вт/(м·К). Материал корпуса должен выбираться исходя как из высокой теплопроводности, так и из его конструктивных свойств, удовлетворяющих воздействующим факторам (прочность, твердость, и др.).

5) В ЭМ, входящих в состав ЭВМ, с установленными на них микропроцессорами с мощностью потребления ~25 Вт, необходимо использовать контурные тепловые трубы (см. рис. 2, 3) с характеристиками приведенными в табл. 3.

6) Для эффективного охлаждения ЭМ в ЭВМ необходимо обеспечить низкую величину контактных тепловых сопротивлений вдоль всего тракта передачи потоков тепла от элект-

тронных модулей и модуля ИЭП к продуваемому через корпус воздух.

К критически важным для уровней температуры микросхем, электронных модулей и модуля ИЭП контактными тепловым сопротивлением, относятся контакты между следующими элементами конструкции рассматриваемой ЭВМ:

- МС микропроцессора и интерфейсом испарителя КТТ (см. рис. 2);
- интерфейсом конденсатора КТТ (планкой) и рамкой ЭМ (см. рис. 3);
- рамкой ЭМ и направляющими пазами (см. рис. 1);
- крышкой корпуса ЭВМ и боковыми стенками (см. рис. 1, рис. 4);
- крышкой корпуса, боковыми стенками и задней стенкой корпуса (см. рис. 1, рис. 4);
- боковыми стенками и задней стенкой корпуса;
- модуля ИЭП и внутренней поверхностью задней стенки, внешняя поверхность которой выполнена в виде штыревого радиатора.

С этой целью необходимо: обеспечить толщину контактных зазоров не более 0,2 мм при высокой чистоте обработки контактных поверхностей, малых значениях неплоскостностей и волнистостей поверхностей; заполнить зазоры между контактирующими поверхностями высокотеплопроводными прокладками (например, фольга из терморасширенного графита (ТРГ)) или высокотеплопроводные пасты (например, Arctic Silver 5); обеспечить достаточно значительные усилия болтового прижатия всех контактирующих поверхностей.

Выполнение приведенных выше рекомендаций по конструкции корпуса ЭВМ приводит к следующим расчетным параметрам теплового и аэродинамического режимов корпуса ЭВМ при нормальном и пониженном давлениях:

#### **нормальное давление воздушной среды:**

– для того, чтобы температура каркаса корпуса была меньше его предельно допустимой температуры (см. п. 3.2)  $T_{ЭВМ, max} = 85^\circ\text{C}$ , необходимо создать массовый расход воздуха  $G_{p, min} \geq 75 \text{ кг/ч}$ , при повышенной рабочей температуре среды  $+55^\circ\text{C}$ . При этом средняя температура воздуха, нагнетаемого через корпус, не будет превосходить  $76^\circ\text{C}$  (см. рис. 5). Для обеспечения возможности «прокачки» массового расхода  $75 \text{ кг/ч}$  и более, при повышенной рабочей температуре среды  $+55^\circ\text{C}$ , необходимо создать разность давлений воздуха между входом и выходом в корпус, удовлетворяющую условию  $\Delta p \geq 100 \text{ Па}$ ;

– температура корпуса ЭВМ будет меньше  $T_{ЭВМ} \leq 75^\circ\text{C}$  (взято с запасом в десять градусов относительно предельно допустимой) при массовом расходе воздуха  $G_{p, min} \geq 125 \text{ кг/ч}$ , при повышенной рабочей температуре среды  $+55^\circ\text{C}$ . При этом для обеспечения возможности «прокачки» массового расхода  $125 \text{ кг/ч}$  и более, при повышенной рабочей температуре среды  $+55^\circ\text{C}$ , необходимо создать разность давлений воздуха между входом и выходом в корпус ЭВМ, удовлетворяющую условию  $\Delta p \geq 250 \text{ Па}$ .

Значения температуры корпуса для температуры окружающей среды  $+55^\circ\text{C}$ ,  $+20^\circ\text{C}$ ,  $-60^\circ\text{C}$  при нормальном давлении приведены на рис. 5.

#### **пониженное давление воздушной среды 2000 Па (15 мм.рт.ст):**

– для того, чтобы температура каркаса корпуса при пониженном давлении и повышенной рабочей температуре среды  $+55^\circ\text{C}$ , была меньше предельно допустимой температуры (п. 3.2)  $T_{ЭВМ, max} = 85^\circ\text{C}$ , необходимо создать массовый расход воздуха  $G_{p, min} \geq 120 \text{ кг/ч}$  (см. рис. 6). При этом средняя температура воздуха, нагнетаемого через корпус, не будет превосходить  $76^\circ\text{C}$ . Для обеспечения возможности «прокачки» массового расхода  $120 \text{ кг/ч}$ , при температуре среды  $+55^\circ\text{C}$ , необходимо создать разность давлений воздуха между входом и выходом в корпус, удовлетворяющую условию  $\Delta p \geq 4,5 \text{ кПа}$ ;

– температура корпуса ЭВМ будет меньше  $T_{ЭВМ} \leq 75^\circ\text{C}$  (взято с запасом в десять градусов относительно предельно допустимой) при массовом расходе воздуха  $G_{p, min} \geq 200 \text{ кг/ч}$  при повышенной рабочей температуре среды  $+55^\circ\text{C}$ . При этом для обеспечения возможности «прокачки» массового расхода  $200 \text{ кг/ч}$  и более, при повышенной рабочей температуре среды  $+55^\circ\text{C}$ , необходимо создать разность давлений воздуха между входом и выходом в корпус, удовлетворяющую условию  $\Delta p \geq 10 \text{ кПа}$ ;

– эффективность охлаждения корпуса ЭВМ при пониженном давлении  $2000 \text{ Па}$  (15 мм.рт.ст.) может быть существенно повышена, если конструкцию корпуса с конвективно-кондуктивным воздушным способом охлаждения изменить на конструкцию, в которой реализуется полностью кондуктивный принцип отвода тепла по массивным боковым стенкам корпуса, прижимаемым к охлаждаемой панели, с заданной температурой.

Значения температуры корпуса для температуры среды  $+55^\circ\text{C}$ ,  $+20^\circ\text{C}$ ,  $-60^\circ\text{C}$  и пониженном давлении, приведены на рис. 6.

Публикация выполнена в рамках государственного задания на проведение фундаментальных научных исследований по программам РАН по теме (проекту) «Разработка общих принципов и методики создания корпусов моноблоков с мощностью потребления до 200 Вт и серверов с мощностью до 2 КВт, обеспечивающих высокоэффективный отвод тепла, основанный на современных технологиях охлаждения с применением тепловых

труб, и обладающих эффективной защитой от электромагнитных излучений, удовлетворяющей техническим требованиям» (0065-2018-0018).

## Modern cooling technology using a loop heat pipes in enclosures of on-board servers with high power consumption

A.V. Bakaldin, A.M. Baranov, P.I. Kandalov, A.G. Madera, O.V. Serdin

**Abstract:** On-board servers handling large data streams operate in extreme environments, under the impact of destabilizing factors and significant power consumption. To ensure the assigned thermal regime of the server, it is necessary to apply modern cooling technologies using loop heat pipes capable of removing the power consumed by the microprocessor to 200 W. In this paper, it is analysed the methods for design of enclosures for servers with application of cooling technologies with the use of loop heat pipes. The thermal mode of the design of the onboard server with ten electronic modules and a power supply block with a total power of 800 W located in a housing that is impervious to penetration of electromagnetic radiation, moisture, dirt and dust, are considered.

**Keywords:** on-board server, computer, cooling, loop heat pipe, thermal modes, heat exchange, microprocessor, enclosure of server, depressed pressure

### Литература

1. С.Г. Бобков, А.Г. Мадера. Энергетические затраты, быстродействие и проблема теплоотвода в микропроцессорах // Программные продукты и системы. 2013. №4. С. 29 – 35.
2. Y. Maysdanik, V. Pastukhov, M. Chernysheva. Development and investigation of a loop heat pipe with a high heat-transfer capacity // Applied Thermal Engineering. 2018. V. 130. P. 1052 – 1061.
3. Ю.Ф. Майданик. Контурные тепловые трубы – высокоэффективные теплопередающие устройства для систем охлаждения электроники // Электроника: Наука, технология, бизнес. 2017. № 6(166). С. 122 – 130.
4. Ю.Ф. Майданик. Контурные тепловые трубы - высокоэффективные теплопередающие устройства, Инновации, 2003, № 5 (62), 83-86.
5. А.Г. Мадера, П.И. Кандалов. Моделирование трехмерных температурных полей в электронных модулях // Программные продукты и системы. 2010. №2. С. 36.
6. А.Г. Мадера Концепция математического и компьютерного моделирования тепловых процессов в электронных системах // Программные продукты и системы. 2015. № 3.С. 79 – 86.
7. А.Г. Мадера Моделирование теплообмена в технических системах. – М.: Науч. фонд «Первая исслед. лаб. им. акад. В.А. Мельникова», 2005.
8. А.В. Бакалдин, П.И. Кандалов, Н.А. Козлов, А.Г. Мадера, О.В. Сердин. Методы обеспечения отвода тепла в корпусах высокопроизводительных защищенных серверов с воздушным охлаждением, Труды НИИСИ РАН, 2018. Т. 8. №4 (см. текущий номер).
9. В.М. Белецкий, Г.А. Кривов. Алюминиевые сплавы (состав, свойства, технология, применение). Справочник / Под общей ред. ак. РАН И.Н. Фридляндера. – Киев: Коминтех, 2005.
10. М.А. Михеев, И.М. Михеева. Основы теплопередачи. – М.: Бастед, 2010.

# Исследование влияния параметров теплового режима на статистические меры стохастической температуры электронного модуля

М.Ж. Акжолов<sup>1</sup>, П.И. Кандалов<sup>2</sup>, А.Г.Мадера<sup>3</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

E-mail's: <sup>1</sup>ak-1@mail.ru, <sup>2</sup>petrki87@gmail.com, <sup>3</sup>alexmadera@mail.ru

**Аннотация.** В статье приведены результаты математического и компьютерного моделирования статистических мер (математических ожиданий и дисперсий) стохастической температуры электронного модуля в корпусе. Получены зависимости статических мер от потребляемой мощности и температуры окружающей среды. Показано, что темп прироста математического ожидания и дисперсии стохастической температуры электронного модуля уменьшается при увеличении величины мощности потребления и температуры окружающей среды.

**Ключевые слова:** электронный модуль, тепловой режим, стохастическая температура, математическое ожидание, дисперсия

В работах [1 – 4, 6, 7] показано, что определяющие тепловой режим электронной системы (ЭС) факторы являются недетерминированными. Так например, значительному разбросу подвержены величины мощностей потребления у различных экземпляров ЭС одного и того же типа, температура окружающей среды, конструктивные параметры ЭС, например, зазоры между элементами и компонентами ЭС, и пр. При этом, определяющие параметры являются интервальными случайными величинами с границами, определяемыми статистическим технологическим разбросом при изготовлении и монтаже компонентов в ЭС. Реальные тепловые процессы, протекающие в ЭС в процессе своего функционирования, характеризуются нестационарностью, нелинейностью (в силу зависимости тепловых параметров ЭС от температуры), интервально стохастическим характером, поэтому для того чтобы модели и методы моделирования тепловых процессов в ЭС были адекватны реальности, они должны учитывать все перечисленные указанные особенности. Созданию таких методов и моделей посвящены работы [1 – 7]. В данной статье получены результаты численного моделирования влияния математических ожиданий таких факторов теплового режима, как мощность потребления ЭС и температура окружающей среды на температуру электронного модуля (ЭМ), заключенного в герметический корпус, конструкция которого подробно приведена в работе [1].

Стохастическая математическая модель, описывающая нестационарные тепловые процессы в тепловой модели рассматриваемой ЭС, имеет следующий вид ( $(t, \omega) \in [0, \tau] \times \Omega$ ) [1]:

$$h_{PCB} \frac{dT_{PCB}(t, \omega)}{dt} + J_{PCB-E}(T_{PCB}, T_E, t, \omega) = \Phi(\omega), \quad (1)$$

$$h_E \frac{dT_E(t, \omega)}{dt} + J_{E-e}(T_E, T_e, t, \omega) - J_{PCB-E}(T_{PCB}, T_E, t, \omega) = 0, \quad (2)$$

где  $T_{PCB} = T_{PCB}(t, \omega)$ ,  $T_E = T_E(t, \omega)$ ,  $T_e = T_e(t, \omega)$  – изотермические интервально стохастические температуры электронного модуля (PCB), корпуса ЭС (E), окружающей среды (e) соответственно;  $h_{PCB} = \rho_{PCB} c_{PCB} V_{PCB}$  – полная теплоемкость электронного модуля объемом  $V_{PCB}$ , плотностью  $\rho_{PCB}$  и удельной теплоемкостью  $c_{PCB}$ ;  $h_E = \rho_E c_E V_E$  – полная теплоемкость конструкции корпуса объемом  $V_E$ , плотностью  $\rho_E$  и удельной теплоемкостью  $c_E$ ;  $\Phi(\omega)$  – мощность потребления электронного модуля;  $J_{PCB-E}(T_{PCB}, T_E, t, \omega) = J_{PCB-E}^{conv} + J_{PCB-E}^{rad}$  – суммарный тепловой поток конвекции и излучения между электронным модулем и корпусом ЭС;  $J_{E-e}(T_E, T_e, t, \omega) = J_{E-e}^{conv} + J_{E-e}^{rad}$  – суммарный тепловой поток конвекции и излучения от корпуса ЭС в окружающую среду.

Уравнения для статистических мер стохастической нестационарной температуры ЭМ в корпусе, полученная в [1], имеет вид:

$$h_{PCB} \frac{d\bar{T}_{PCB}(t, \omega)}{dt} + \bar{J}_{PCB} = \bar{\Phi}, \quad (3)$$

$$h_{PCB} \frac{dVar_{TPCB}(t)}{dt} + 2f_1 Var_{TPCB}(t) + 2f_2 K_{TPCB-\Phi}(t) - 2K_{TPCB-\Phi} = 0, \quad (4)$$

с начальными условиями:

$$\bar{T}_{PCB}(t=0) = \bar{T}_E(t=0) = \bar{T}_e,$$

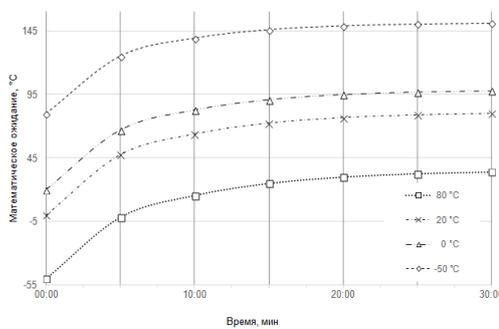
$$Var_{TPCB}(t=0) = Var_{TE}(t=0) = Var_{Te},$$

$$K_{TPCB-TE}(t=0) = Var_{Te},$$

$$K_{TPCB-\Phi}(t=0) = K_{TE-\Phi}(t=0) = 0,$$

$$K_{TPCB-Te}(t=0) = K_{TE-Te}(t=0) = Var_{Te},$$

где  $Var_{TPCB}(t)$ ,  $Var_{TE}(t)$  – дисперсии температур электронного модуля и корпуса ЭС;



Результаты компьютерного моделирования математических ожиданий (МО) и дисперсий (Д) стохастической нестационарной температуры ЭМ при различных температурах окружающей среды  $T_a$  приведены на рис. 1. Начальные условия соответствовали температуре окружающей  $\bar{T}_a^0$  равной  $-50, 0, 20$  и  $80$  °C, потребляемая ЭМ мощность  $\Phi_u$  составляла 30 Вт.

Установившиеся значения математического ожидания стохастической температуры ЭМ  $\bar{T}_{ЭМ}$  достигались через время, равное 30 мин и составили 151.0, 97.7, 79.7 и 34.0 °C при соответствующих значениях температуры окружающей среды  $\bar{T}_a$ . Дисперсии стохастической температуры ЭМ составили  $3.8^{\circ}\text{C}^2$

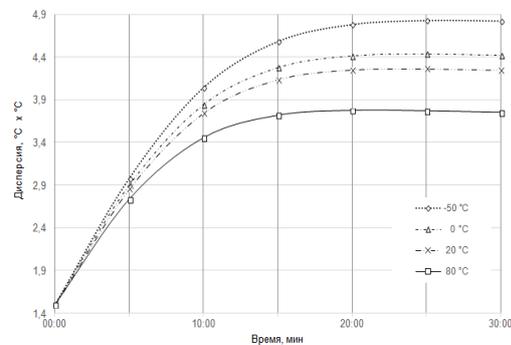


Рис. 1. Нестационарные МО (°C) и Д (°C²) стохастической температуры ЭМ при различных температурах среды  $T_a$

$K_{TPCB-TE}(t)$  – ковариация между стохастическими температурами  $T_{PCB}(t, \omega)$  и  $T_E(t, \omega)$ ;  $Var_{Te}$  – дисперсия стохастической температуры среды;  $Var_{\Phi}$  – дисперсия стохастической мощности тепловыделения электронного модуля;  $K_{TPCB-\Phi}$ ,  $K_{TE-\Phi}$  – ковариации между стохастическими температурами электронного модуля и корпуса ЭС и стохастической мощностью тепловыделения электронного модуля;  $K_{TPCB-Te}$ ,  $K_{TE-Te}$  – ковариации между стохастическими температурами электронного модуля и корпуса ЭС и стохастической температурой окружающей среды. В уравнениях (3), (4) все входящие в них величины и функции являются известными из исходных данных или легко вычисляются.

Для решение системы дифференциальных уравнение (3), (4) [1] применялся численный метод решения задачи Коши для обыкновенных дифференциальных уравнений – Рунге-Кутты 4-го порядка [8]. Для величины шага  $t=10^{-2}$  время счета составляла не более 1с.

при  $\bar{T}_{ЭМ} = 150.95$  °C,  $4.3^{\circ}\text{C}^2$  при  $\bar{T}_{ЭМ} = 97.70$  °C,  $4.41^{\circ}\text{C}^2$  при  $\bar{T}_{ЭМ} = 79.69$  °C и  $4.82^{\circ}\text{C}^2$  при  $\bar{T}_{ЭМ} = 34.00$  °C. Результаты расчетов показывают, что темп прироста дисперсии снижается по мере роста температуры окружающей среды  $T_a$ , поскольку при меньших температурах окружающей среды, градиент температуры вокруг ЭМ больше, т.е. конвекция развивается намного интенсивнее, а по мере возрастания температуры окружающей среды  $\bar{T}_a$  значения градиента уменьшаются и влияние конвекционной составляющей на температуру электронного модуля ослабевает, приводя к снижению разброса стохастической температуры ЭМ.

Результаты компьютерного моделирования зависимостей математического ожидания и дисперсии стохастической температуры ЭМ при различных значениях мощности потребления ЭМ  $\bar{\Phi}_u$ , равных 10, 20, 30, 50, 100, 500 и 1000 Вт, приведены на рис. 2. Установившиеся значения МО стохастической температуры электронного модуля  $\bar{T}_{ЭМ}$  в момент

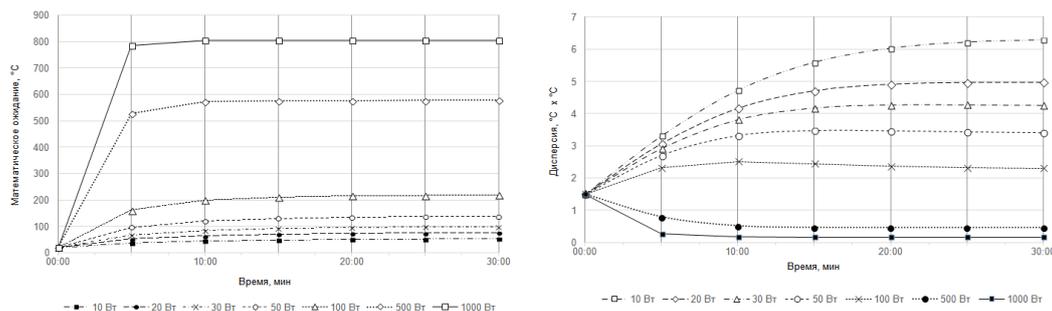


Рис. 2. Нестационарные МО ( $^{\circ}\text{C}$ ) и Д ( $^{\circ}\text{C}^2$ ) стохастической температуры ЭМ при различных мощностях потребления ЭМ  $\bar{\Phi}_n$

времени  $t = 30$  мин составили соответственно: 51.5, 76.0, 97.9, 136.7, 217.4, 577.4, 805.1,  $^{\circ}\text{C}$ , а дисперсии — 6.3, 5.0, 4.2, 3.4, 2.3, 0.45, 0.16,  $^{\circ}\text{C}^2$ . Сравнение полученных зависимостей между собой показывает, что установившийся режим для МО достигается через 10 мин, а для Д — через  $t = 30$  мин. С ростом мощности потребления ЭМ, приращение величины дисперсии стохастической температуры ЭМ падает, что обуславливается тем, что по мере увеличения потребляемой мощности влияние разброса как температуры сре-

ды, так и собственной мощности (величина разброса мощности задана постоянной и не зависящей от ее МО) уменьшается.

Публикация выполнена по госзаданию ФГУ ФНЦ НИИСИ РАН по теме 0065-2018-0003 «Исследование и разработка моделей и методов математического и компьютерного моделирования тепловых процессов в электронных системах различного назначения в условиях неопределенности» АААА-А18-118032290204-2.

## Study of the influence of thermal parameters on the statistical measures of the stochastic temperature of the electronic module

M.J. Akjolov, P.I. Kandalov, A.G. Madera

**Abstract.** The article presents the results of mathematical and computer modeling of statistical measures (expectations and variances) of the stochastic temperature of the electronic module in the case. The dependences of static measures on power consumption and ambient temperature are obtained. It is shown that the growth rate of the expectation and dispersion of the stochastic temperature of the electronic module decreases with increasing power consumption and ambient temperature.

**Keywords:** electronic module, thermal mode, stochastic temperature, expectation, variance

## Литература

1. А.Г. Мадера, П.И. Кандалов. Математическое моделирование интервально стохастических тепловых процессов в технических системах при интервальной неопределенности определяющих параметров // Компьютерное исследование и моделирование. 2016. Т. 8. №3. С. 501 – 520.
2. П.И. Кандалов. Интервально стохастические тепловые процессы, протекающие в электронном модуле, заключенном в герметичном корпусе // Труды НИИСИРАН. 2016. Т. 6. №1. С. 64 – 68.
3. А.Г. Мадера, П.И. Кандалов. Моделирование трехмерных температурных полей в электронных модулях // Программные продукты и системы. 2010. №2. С. 36
4. А.Г. Мадера, П.И. Кандалов. Моделирование температурных полей технических систем в условиях интервальной неопределенности // Тепловые процессы в технике. 2014. № 5. С. 225 – 229
5. J.C. Georgiadis. On the approximate solution on non-deterministic heat and mass transport problems // Int. J. Heat Mass Transfer. 1991. V. 33. n. 8. P.2099 – 2105
6. C.J. Keller, V.W. Antonetti Statistical thermal design for computer electronics // Electronic Packaging and Production. 1979. V.19.n.3. P. 55 – 62.
7. A.G. Madera. Simulation of stochastic heat conduction processes // International Journal of Heat and Mass Transfer. 1994. Т. 37. № 16. С. 2571 – 2577.
8. Н.С. Бахвалов, Н.П. Жидков., Г.М. Кобельков. Численные методы: Учеб. пособие. – М.: Наука. Гл. ред. физ.-мат. лит., 1987. – 600 с.

# Тепловой режим микросхемы в условиях неопределенности определяющих параметров и воздействии тепловой обратной связи

А.Г. Мадера

*ФГУ ФНЦ НИИСИ РАН, Москва, Россия, E-mail: omm@niisi.ras.ru,*

**Аннотация:** Рассматривается тепловой режим интегральной полупроводниковой микросхемы в условиях неопределенности значений параметров теплового режима микросхем и воздействии тепловой обратной связи. Метод анализа основан на решении уравнения математической макромодели микросхемы относительно усредненной стохастической температуры кристалла и полученных аналитических выражений для математического ожидания и дисперсии температуры. Показано, что совместное воздействие тепловой обратной связи и интервально-стохастической неопределенности параметров микросхемы оказывает значительное влияние на температуру микросхемы, а пренебрежение указанными факторами может приводить к значительным ошибкам при проектировании микросхем и электронных устройств на их основе.

**Ключевые слова:** микросхема, тепловой режим, тепловая обратная связь, температура, неопределенность, математическое ожидание, дисперсия

Явление тепловой обратной связи (ТОС), возникающее в процессе функционирования электронных систем (ЭС) и интегральных полупроводниковых микросхемах, и ее влияние на тепловой и электрический режимы МС, подробно рассмотрено в работах [1, 2, 3, 4]. Природа возникновения ТОС в электронных устройствах основанных на полупроводниковой материалах и технологии, обуславливается взаимозависимостью электрических параметров МС от температуры и температуры МС от мощности потребления, определяемой электрическими параметрами. Явление ТОС в МС возникает в процессе ее функционирования и формируется следующим образом: потребляемая МС мощность переходит в тепло, обуславливая установление в ней температурного поля, которое, воздействуя на мощность потребления, в силу ее значительной зависимости от температуры, приводит к ее изменению и, как следствие, к изменению температурного поля, замыкая тем самым петлю тепловой обратной связи. Таким образом эффект ТОС заключается во взаимообусловленном взаимодействии и взаимовлиянии между собой изменений температуры и мощности потребления в МС, то есть взаимовлиянии факторов и параметров теплового (мощность разогрева, температуры в различных точках конструкции МС, температурно-влажностные параметры окружающей среды) и электрического (мощность потребления, быстродействие, помехоустойчивость, и пр.) режимов друг на друга.

Воздействие ТОС на электрический и тепловой режимы в МС, а также в электронных системах (ЭС) с применением МС, обуславливает выход электрических (статических и динамических) параметров МС за пределы допустимых значений, снижение надежности, работоспособности, помехоустойчивости, повышенные уровни температуры, и др. Влияние ТОС на электрический режим МС и ЭС присутствует всегда и является неустранимым, хотя в ряде случаев и может быть уменьшено, например, за счет применения более эффективных способов охлаждения МС. Пренебрежение воздействием ТОС при проектировании электрических и тепловых режимов МС, равно как и ЭС с их применением, приводит к ошибкам их проектирования и созданию неконкурентоспособных как МС, так и ЭС на их основе.

Между тем, в процессе функционирования МС реальные параметры ее теплового и электрического режимов определяются не только воздействием ТОС, но и являются, как показано в [5], неопределенными, а именно, – интервально-стохастическими. Неопределенный характер определяющих параметров теплового и электрического режимов МС обуславливается такими факторами, как статистический технологический разброс как при изготовлении, так и монтаже МС и ЭС; интервально-стохастический разброс тепловых и электрических параметров МС, которые возникают только в процессе функционирования МС и ЭС; интервально-стохастический разброс параметров

окружающей среды. Интервально-стохастический характер параметров МС обуславливается тем, что реальные параметры различных образцов МС в партии, состоящих из таких же «идентичных» МС, характеризуются тем, что изменяются от образца к образцу, оставаясь при этом внутри некоторого интервала своих значений,

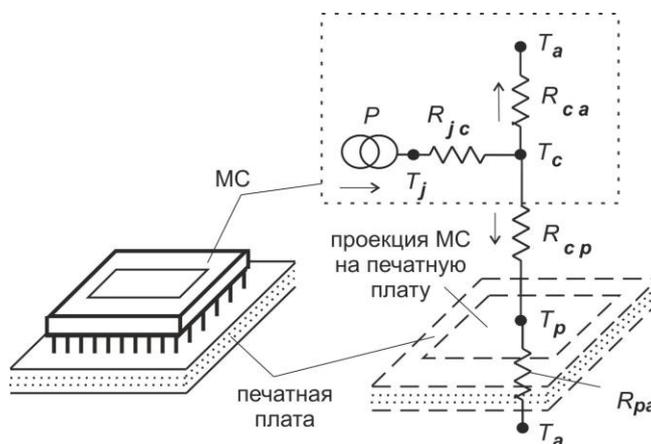


Рис. 1. ЭС, содержащая МС припаянную выводами корпуса к печатной плате, тепловая схема ЭС. Пунктиром обведена тепловая схема МС

границы которого определяются общим разбросом параметров по всей партии МС. Какое именно значение имеет тот или иной параметр в конкретном экземпляре МС неизвестно (невозможно измерять все параметры всех МС в партии), однако, границы его допустимых значений, обусловленных статистическим технологическим разбросом при изготовлении МС, могут быть приближенно установлены. В работах [5] установлено, что адекватной математической моделью неопределенного параметра является модель параметра в виде равномерно распределенной случайной величины, ограниченной интервалом разброса.

Оба фактора, – ТОС и интервально-стохастическая неопределенность параметров теплового и электрического режимов, – оказывают значительное влияние на реальные электрические и тепловые параметры МС в процессе их функционирования и не устранимы при любых технологиях изготовления, монтажа и режимах эксплуатации.

В статье проводится анализ влияния ТОС и неопределенности на значения параметров теплового режима МС как по отдельности, так и совместно. Анализ основан на математической макромоделе МС, описывающей температуру кристалла МС, заключенной в корпус, установленной на печатной плате и находящейся в условиях теплообмена с окружающей средой.

Рассмотрим ЭС, состоящую из МС, заключенной в корпус, припаянном своими выводами к печатной плате (рис. 1) и ее тепловую модель.

Тепловая модель ЭС (рис. 1) построена на основании принципа электротепловой аналогии и представляет собой тепловую схему в виде направленного графа [6],

пунктиром обведена тепловая схема МС. Узлы тепловой схемы моделируют физические компоненты ЭС (МС, печатную плату и окружающую среду) в изотермическом приближении, а ветви, с включенными в них тепловыми сопротивлениями, – определенный вид теплообмена между компонентами ЭС. узлы моделируют кристалл МС с температурой  $T_j$ , корпус МС с температурой  $T_c$ , печатную плату с температурой  $T_p$  и окружающую среду с температурой  $T_a$ . Тепловые сопротивления в ветвях графа моделируют теплообмен между компонентами ЭС, а именно: тепловое сопротивление  $R_{jc}$  моделирует кондуктивный перенос тепла от кристалла МС к внешней поверхности корпуса (тепловое сопротивление корпуса), тепловое сопротивление  $R_{cp}$  моделирует кондуктивный перенос тепла через зазор между нижней поверхностью корпуса МС и печатной платой, тепловые сопротивления  $R_{ca}$  и  $R_{pa}$  моделируют конвективный теплообмен с поверхности корпуса и печатной платы в среду, соответственно, элемент независимого источника моделирует тепловыделение на кристалле МС мощностью  $P$ .

В реальных условиях функционирования параметры теплового и электрического режимов ЭС находятся под воздействием ТОС, что выражается в виде зависимости потребляемой МС мощности  $P(T_j)$  от

температуры кристалла  $T_j$ , и кроме того, определяющие тепловой режим ЭС параметры, такие как тепловые сопротивления  $R_{jc}(\omega)$ ,  $R_{cp}(\omega)$ , температура среды  $T_a(\omega)$ , а также мощность потребления  $P(T_j, \omega)$ , являются интервально-стохастическими, то есть функциями элементарных событий  $\omega$  из пространства элементарных событий  $\Omega$ .

Математическая интервально-стохастическая макромодель рассматриваемой ЭС для каждого  $\omega \in \Omega$  может быть записана в следующем виде:

$$T_j(\omega) = T_a(\omega) + (R_{jc}(\omega) + R_{c-p-a}(\omega)) \cdot P(T_j, \omega), \quad (1)$$

где  $P(T_j, \omega)$  – интервально-стохастическая мощность, линейно зависящая от температуры кристалла МС

$$P(T_j, \omega) = P_0(\omega) + \alpha(\omega) \cdot T_j(\omega); \quad (2)$$

$P_0(\omega)$  и  $\alpha(\omega)$  – интервально-стохастические величины мощности при температуре кристалла  $T_{j0}$  и коэффициента ТОС, Вт/К;  $R_{c-p-a}(\omega)$  – интервально-стохастическая величина теплового сопротивления между корпусом МС и окружающей средой, равная параллельному соединению тепловых сопротивлений  $R_{ca}$  и  $R_{cp}(\omega) + R_{pa}$

$$R_{c-p-a}(\omega) = \frac{R_{ca} (R_{cp}(\omega) + R_{pa})}{R_{ca} + R_{cp}(\omega) + R_{pa}}$$

Интервально-стохастический характер теплового сопротивления  $R_{cp}(\omega)$  обусловлен статистическим технологическим разбросом толщина зазора между корпусом МС и печатной платой (рис. 1). В данной статье принимается, что толщина зазора между корпусом МС и печатной платой является фиксированной и детерминированной, так что тепловое сопротивление зазора  $R_{cp}$  также будет детерминированным, иначе говоря, статистический технологический разброс при монтаже МС на печатной плате в партии всех идентичных образцов ЭС не учитывается. Коэффициент интенсивности ТОС  $\alpha(\omega)$  в выражении для мощности потребления  $P(T_j, \omega)$  также является общим случае интервально-стохастической величиной, обусловленной статистическим технологическим разбросом изготовления МС, который, однако, как и в случае с тепловым сопротивлением  $R_{cp}(\omega)$ , будем считать детерминированным. Остальные случайные величины в математической модели (1), (2) считаются независимыми, что, в принципе, соответствует практике.

С учетом приведенных замечаний, интервально-стохастическая математическая модель ЭС (1), (2) примет вид:

$$T_j(\omega) = T_a(\omega) + (R_{jc}(\omega) + R_{c-p-a}) \cdot P(T_j, \omega), \quad (3)$$

$$P(T_j, \omega) = P_0(\omega) + \alpha \cdot T_j(\omega), \quad (4)$$

где

$$R_{c-p-a} = \frac{R_{ca} (R_{cp} + R_{pa})}{R_{ca} + R_{cp} + R_{pa}}.$$

Введя новую переменную  $\Delta T_j(\omega) = T_j(\omega) - T_a(\omega)$ , равную превышению температуры кристалла МС  $T_j(\omega)$  над температурой окружающей среды  $T_a(\omega)$ , из соотношений (3), (4) получим выражение для интервально-стохастической температуры  $\Delta T_j(\omega)$ , учитывающее как эффект ТОС, так и статистический технологический разброс при изготовлении МС

$$\Delta T_j(\omega) = \frac{R_{jc}(\omega) + R_{c-p-a}}{1 - \alpha \cdot (R_{jc}(\omega) + R_{c-p-a})} \cdot P_0(\omega).$$

Отметим, что как следует из последнего выражения, интервально-стохастический характер температуры кристалла МС  $T_j(\omega)$  обуславливается статистическим технологическим разбросом изготовления МС вызывающим интервально-стохастический характер как теплового сопротивления корпуса МС  $R_{jc}(\omega)$ , так и потребляемой МС мощности  $P_0(\omega)$  при температуре кристалла МС  $T_{j0}$ .

В силу интервально-стохастического характера математической модели неопределенности тепловое сопротивление  $R_{jc}(\omega)$  и потребляемая мощность  $P_0(\omega)$  являются равномерно распределенными случайными величинами с плотностью вероятностей  $p_{R_{jc}} = 1/\Delta_{R_{jc}}$  и  $p_{P_0} = 1/\Delta_{P_0}$  определенных в интервалах  $\Delta_{R_{jc}}$  и  $\Delta_{P_0}$  равных  $\Delta_{R_{jc}} = R_{jc}^B - R_{jc}^H$  и  $\Delta_{P_0} = P_0^B - P_0^H$  соответственно, где  $R_{jc}^H$ ,  $R_{jc}^B$  и  $P_0^H$ ,  $P_0^B$  – нижние и верхние границы интервалов  $\Delta_{R_{jc}}$  и  $\Delta_{P_0}$ .

Статистические меры интервально-стохастической температуры  $\Delta T_j(\omega)$  – математическое ожидание  $\Delta \bar{T}_j = E\{\Delta T_j(\omega)\}$  и дисперсия  $D_{\Delta T_j} = E\{(\Delta T_j(\omega) - \Delta \bar{T}_j)^2\}$  – будут равны соответственно:

$$\Delta \bar{T}_j = E \left\{ \frac{R_{jc}(\omega) + R_{c-p-a}}{1 - \alpha \cdot (R_{jc}(\omega) + R_{c-p-a})} \right\} \cdot \bar{P}_0,$$

$$D_{\Delta T_j} = E \left\{ \frac{(R_{jc}(\omega) + R_{c-p-a})^2}{[1 - \alpha \cdot (R_{jc}(\omega) + R_{c-p-a})]^2} \right\} \cdot E\{P_0^2(\omega)\} - \Delta \bar{T}_j^2,$$

где  $E\{\cdot\}$  – оператор математического ожидания;  $\bar{P}_0$  – математическое ожидание мощности потребления  $P_0(\omega)$ .

Введя обозначения  $A = 1 - \alpha \cdot (R_{jc} + R_{c-p-a})$  и  $B = 1 - \alpha \cdot (\bar{R}_{jc} + R_{c-p-a})$  и вычисляя операторы математического ожидания  $E\{\cdot\}$  для математического ожидания и дисперсии, после несложных преобразований получим:

температуры кристалла МС  $\Delta T_j$  в зависимости от коэффициента ТОС  $\alpha$ , при совместном воздействии ТОС и интервально-стохастической неопределенности, приведена на рис. 2. Анализ полученных зависимостей показывает, что совместное влияние ТОС и

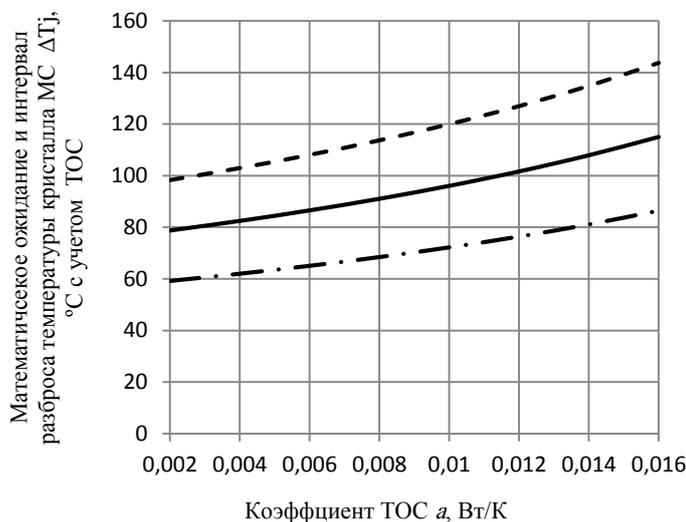


Рис. 2. Зависимость температуры кристалла МС  $\Delta T_j$  от коэффициента ТОС  $\alpha$  при совместном воздействии ТОС и интервально-стохастической неопределенности

$$\Delta \bar{T}_j = \frac{1}{\Delta R_{jc}} \left( -\frac{\Delta R_{jc}}{\alpha} + \frac{1}{\alpha^2} \ln \frac{A}{B} \right) \cdot \bar{P}_0,$$

$$D_{\Delta T_j} = \frac{1}{\alpha^3 \Delta R_{jc}} \left( \alpha \cdot \Delta R_{jc} - 2 \ln \frac{A}{B} + \frac{\alpha \cdot \Delta R_{jc}}{A \cdot B} \right) (D_{P_0} + \bar{P}_0^2) - \Delta \bar{T}_j^2,$$

где  $D_{P_0}$  – дисперсия мощности потребления  $P_0(\omega)$ .

Вычисленные величины математического ожидания  $\Delta \bar{T}_j$  и дисперсии  $D_{\Delta T_j}$  позволяют определить интервал, в котором будут содержаться реальные значения температуры кристалла МС. Отметим, что указанный интервал учитывает воздействие ТОС и интервально-стохастическую неопределенность параметров МС – мощности потребления и теплового сопротивления корпуса. Нижняя и верхняя границы интервала значений температуры кристалла МС определяются по формулам:

$$\Delta T_j^H = \Delta \bar{T}_j - 3 \cdot \sigma_{\Delta T_j}, \quad \Delta T_j^B = \Delta \bar{T}_j + 3 \cdot \sigma_{\Delta T_j},$$

где  $\sigma_{\Delta T_j} = \sqrt{D_{\Delta T_j}}$  – среднее квадратическое отклонение превышения температуры кристалла МС  $T_j(\omega)$  над температурой окружающей среды  $T_a(\omega)$ .

Расчитанные математическое ожидание и границы интервала реальных значений

условий интервальной неопределенности на температуру кристалла МС является значительным.

Так, математическое ожидание температуры кристалла МС  $\Delta \bar{T}_j$  возрастает по сравнению с температурой  $\Delta \bar{T}_j$ , рассчитанной в пренебрежении ТОС ( $\alpha = 0$ ) на 46%, при этом коэффициент интенсивности ТОС увеличивается от  $\alpha = 0$  до  $\alpha = 0,016$ , Вт/К.

Что касается интервала  $\Delta_{\Delta T_j} = \Delta T_j^B - \Delta T_j^H$ , в котором будут находиться реальные значения температуры кристалла МС  $\Delta T_j$ , то его размах возрастает на 47% с увеличением коэффициента интенсивности ТОС  $\alpha$  по отношению к интервалу температуры, рассчитанному в пренебрежении воздействием ТОС ( $\alpha = 0$ ).

Таким образом, совместное воздействие ТОС и интервально-стохастической неопределенности параметров МС, оказывает значительное влияние на тепловой режим МС и им пренебрегать нельзя.

Публикация выполнена по госзаданию ФГУ ФНЦ НИИСИ РАН по теме 0065-2018-0003 «Исследование и разработка моделей и методов математического и компьютерного моделирования тепловых процессов в электронных системах различного назначения в условиях неопределенности» АААА-А18-118032290204-2.

# The thermal mode of the chip in conditions of uncertainty of the determining parameters and the effects of thermal feedback

A.G. Madera

**Annotation:** The thermal mode of an integrated circuit is considered under conditions of uncertainty of the values of the parameters of the thermal mode of the integrated circuit and the effect of thermal feedback. The analysis method is based on solving the equation of a mathematical macro model of a integrated circuit with respect to the average stochastic temperature of the chip and the resulting analytical expressions for the expectation and dispersion of the temperature. It is shown that the combined effect of thermal feedback and interval-stochastic uncertainty of the integrated circuit parameters has a significant impact on its temperature and neglecting these factors can lead to significant errors in designing chips and electronic devices based on them.

**Keywords:** integrated circuit, thermal mode, thermal feedback, temperature, uncertainty, mathematical expectation, variance

## Литература

1. А.Г. Мадера. Воздействие тепловой обратной связи на температуру кристалла микросхемы // Труды научно-исследовательского института системных исследований Российской академии наук. 2018. Т. 8. №2. С. 26-28.
2. А.Г. Мадера. Тепловые процессы в электронных системах в условиях влияния эффекта тепловой обратной связи // Тепловые процессы в технике. 2018. Т. 10. № 3-4. С. 144-151.
3. А.Г. Мадера. Моделирование воздействия тепловой обратной связи на тепловые процессы в электронных системах // Компьютерные исследования и моделирование. 2018. Т. 10. №4. С. 483-494.
4. В.А. Сергеев, А.М. Ходаков. Нелинейные тепловые модели полупроводниковых приборов. – Ульяновск: УлГТУ, 2012.
5. A.G. Madera. Interval-Stochastic Thermal Processes in Electronic Systems: Analysis and Modeling // Journal of Engineering Thermophysics. 2017. V. 26. N. 1. P. 17-28.
6. А.Г. Мадера. Моделирование теплообмена в технических системах. – М.: Научный Фонд «ПИЛ им. акад. В.А. Мельникова», 2005.

# Метод стохастического псевдорешения при математическом моделировании тепловых режимов в электронных системах в условиях неопределенности определяющих параметров

А.Г. Мадера

ФГУ ФНЦ НИИСИ РАН, Москва, Россия, E-mail: omm@niisi.ras.ru

**Аннотация:** Предложен метод математического моделирования стохастических тепловых режимов электронных систем, основанный на понятиях псевдорешения и псевдообратной матрицы. В отличие от существующих разработанный метод является универсальным, не ограничен видом функций распределения вероятностей стохастических параметров электронной системы и окружающей среды. Получены матричные уравнения для статистических мер стохастических температур элементов тепловой модели электронной системы, а именно, вектора математических ожиданий и ковариационной матрицы.

**Ключевые слова:** тепловой режим, электронная система, псевдорешение, неопределенность, статистические меры, математическое ожидание, дисперсия

## Введение

Реальные тепловые процессы, протекающие при функционировании электронных систем (ЭС), как показано в работах [6 – 10, 14 – 16], являются стохастическими. Иными словами, параметры определяющие тепловые процессы в ЭС являются не детерминированными, а интервально стохастическими, поскольку подвержены значительному статистическому разбросу внутри своих интервалов изменения. Поэтому распределения температуры в различных элементах ЭС и тепловые процессы в ЭС будут в реальной действительности носить интервально-стохастический характер.

Интервально стохастическая неопределенность параметров и факторов, определяющих интервально стохастические тепловые процессы в ЭС, обуславливается такими интервально стохастическими факторами, как: (1) статистическим технологическим разбросом параметров компонентов ЭС при их изготовлении, сборке и монтаже в ЭС; (2) случайными факторами, обусловленными функционированием ЭС (токи и напряжения электропитания, мощности потребления интегральных микросхем (ИС) и электро- радио элементов (ЭРЭ), значения температуры и скорости потоков охлаждающей среды внутри ЭС; (3) случайностью параметров окружающей среды (температура, давление, скорости потоков). Учет стохастичности параметров, определяющих тепловой режим в ЭС позволяет адекватно рассчитывать распределения температуры ЭС и ее элементов.

Стохастический характер неопределенности факторов и параметров, определяющих тепловые процессы в ЭС, является неустрашимым и имеет место при любой технологии изготовления и монтажа элементов ЭС. Поэтому проблема математического и компьютерного моделирования интервально стохастических тепловых процессов при воздействии интервально стохастических факторов и параметров является актуальной для адекватного теплового проектирования ЭС. Между тем, несмотря на важность и актуальность этой проблемы, работ, посвященных указанной проблематике явно недостаточно, за исключением работ, посвященных моделированию стохастических температурных полей в телах простейшей формы, что обуславливается чрезвычайной сложностью математического моделирования стохастических температурных распределений в сколь-нибудь сложных системах.

Существующие методы носят, в основном, характер *ad hoc* применительно к конкретному случаю, что не позволяет использовать их на практике при математическом / компьютерном моделировании и тепловом проектировании сложных ЭС. Применение же общих и универсальных методов, таких как метод статистических испытаний [5], методы стохастического оператора [Адомиан], стохастической обратной матрицы [6] и разложения нелинейностей в ряд Тейлора [10, 15, 16] наталкиваются для получения адекватных результатов моделирования на непреодолимые трудности, обусловленные как сложностью математической модели, описывающей стохастические температурные распределе-

ния в сложных системах, которая представляет собой систему стохастических нестационарных и нелинейных уравнений в частных производных, так и значительными затратами машинного времени и памяти даже с применением супер-ЭВМ.

Анализ стохастических тепловых процессов даже в линейном случае связан со значительными трудностями, вызванными тем, что в явном виде определить статистические меры стохастического вектора температур  $T(\omega)$ , определяемого обобщенным матрично-стохастическим уравнением  $H(\omega)T(\omega) = Y(\omega)$ , (вектор математических ожиданий  $E\{T(\omega)\} = \bar{T}$ , дисперсии  $E\{(T(\omega) - \bar{T})^2\} = D_T$ , среднеквадратические отклонения  $\sigma_T = \sqrt{D_T}$ , ковариационную матрицу  $K_{TT} = E\{T(\omega)T^T(\omega)\}$ , где  $(\cdot)^T$  – операция транспонирования матрицы), не представляется возможным.

Действительно, воздействие оператора математического ожидания  $E\{\cdot\}$  на левую часть уравнения  $H(\omega)T(\omega) = Y(\omega)$  не позволяет получить выражение для математического ожидания  $\bar{T}$  стохастической температуры  $T(\omega)$ , поскольку  $E\{H(\omega)T(\omega)\} \neq E\{H(\omega)\} \cdot E\{T(\omega)\}$ , поскольку стохастическая температура  $T(\omega)$  и стохастическая матрица  $H(\omega)$  являются зависимыми стохастическими величинами. Попытки найти выражение  $T(\omega)$  через обратную стохастическую матрицу  $H^{-1}(\omega)$ , а именно, как  $T(\omega) = H^{-1}(\omega)Y(\omega)$  также не приводят к результату, поскольку обратную стохастическую матрицу  $Y(\omega)$  невозможно определить в явном виде, а ее поиск для всех реализаций  $\omega \in \Omega$  наталкивается на значительные вычислительные трудности и занимает много машинного времени (методом статистических испытаний). В работах [1, 6] предложен метод, который, в принципе, позволяет находить статистические меры стохастического вектора температур  $T(\omega)$  графа тепловой модели. А именно, стохастическая матрица  $H(\omega)$  представляется в виде суммы матриц  $H(\omega) = \bar{H} + H^0(\omega)$ , где  $\bar{H}$  – математическое ожидание матрицы  $H(\omega)$ ,  $H^0(\omega)$  – стохастическая матрица с нулевым математическим ожиданием. Представив матрицу  $H(\omega)$  в виде  $H(\omega) = \bar{H}(I + \bar{H}^{-1}H^0(\omega))$  получим выражение для обратной стохастической матрицы  $H^{-1}(\omega)$  в виде бесконечного матричного ряда

$$\begin{aligned} H^{-1}(\omega) &= (I + \bar{H}^{-1}H^0(\omega))^{-1} \cdot \bar{H}^{-1} = \\ &= \sum_{k=1}^{\infty} (-1)^k (\bar{H}^{-1}H^0(\omega))^k \cdot \bar{H}^{-1}, \end{aligned}$$

сходящегося при условии на матричную норму  $\|\bar{H}^{-1}H^0(\omega)\| < 1$ .

В таком представлении обратной матрицы  $H^{-1}(\omega)$  появляется возможность получить явный вид математического ожидания вектора  $T(\omega)$ , а именно:

$$T(\omega) = \sum_{k=1}^{\infty} (-1)^k E\{(\bar{H}^{-1}H^0(\omega))^k \bar{H}^{-1} \cdot Y(\omega)\}.$$

Учитывая независимость элементов стохастической матрицы  $H^0(\omega)$  и вектора  $Y(\omega)$  можно получить явные выражения для статистических мер. Но и в этом случае, в связи с техническими трудностями, получить приемлемое выражение возможно лишь ограничившись членами ряда со степенью не выше второй, что во многих случаях не позволяет достичь приемлемой точности.

В статье предлагается метод псевдорешения для математического моделирования стационарных интервально стохастических тепловых режимов в ЭС, позволяет с достаточной для практики точностью рассчитывать интервальные статистические меры (математические ожидания, дисперсии, среднеквадратические отклонения ковариации) параметров теплового режима в ЭС при условии задания статистических мер входных интервально-стохастических определяющих параметров. Метод легко реализуется в виде компьютерных программ для проведения численных расчетов на современных персональных компьютерах и супер-ЭВМ.

## Тепловая и математическая модели интервально стохастических тепловых процессов в ЭС

Тепловая модель ЭС представляет собой систему  $N$  изотермических элементов [6, 13], которая получается после дискретизации конструкции ЭС на составляющие элементы до размеров элементов, при которых с достаточной для практики точностью можно принять, что распределение температуры в них является изотермическим, в том числе и температура *априори* неизвестной жидкостной среды внутри корпуса. Тепловая схема соответствующая тепловой модели ЭС, построенная на основании электротепловой аналогии [6, 13], представляет собой граф, состоящий из  $N$  узлов,  $M$  ветвей,  $N - 2$  независимыми источниками тепловых потоков ( $\Phi_i$ ,  $i = 1, 2, \dots, N - 2$ ) и двумя узлами с *априори* известными и заданными температурой окружающей среды  $T_e$  и температурой жидкостного потока на входе в ЭС  $T_{a,in}$ . Согласно электротепловой аналогии, узлы тепловой схемы моделируют изотермические элементы тепловой модели; потенциалы в узлах – температуры изотерми-

ческих элементов; токи в ветвях – тепловые потоки; независимые источники токов – источники теплоты (мощностей потребления) активных элементов; независимые потенциалы в узлах – заданные температуры; ветви с конденсаторами – полные объемные теплоемкости элементов тепловой модели.

Математическая модель, описывающая интервально стохастические тепловые процессы в ЭС строится на основе графа тепловой схемы.

Тепловой процесс в ЭС однозначно определяется такими факторами, как: форма, размеры и пространственное размещение компонентов в ЭС и конструкцией ЭС; мощности потребления активных компонентов; граничные условия на поверхностях компонентов и корпуса ЭС; характер теплообмена, определяемый конвекцией (вынужденной или естественной), излучением и кондукцией; температура и скорость жидкостной среды внутри корпуса ЭС; температура окружающей среды; начальные условия в начальный момент времени; теплофизические характеристики (плотность, теплопроводность, теплоемкость, степень черноты излучения) материалов элементов, жидкостной и окружающей среды.

Если все факторы и параметры, определяющие тепловые процессы в ЭС, являются детерминированными, то математическая модель теплового процесса также будет детерминированной.

В случае, если хотя бы один из параметров или факторов теплового процесса является интервально стохастическим, то и сам тепловой процесс в ЭС будет носить интервально стохастический характер и описываться стохастической математической моделью.

Стохастическая математическая модель, описывающая интервально стохастические стационарные температуры  $T_i(\omega)$ ,  $i = 1, 2, \dots, N$ , элементов тепловой схемы (рис. 2), имеет вид [6]

$$AG(\omega)A^T \cdot T(\omega) = \Phi(\omega) + AG(\omega) \cdot T_a(\omega), \quad (1)$$

где  $T(\omega) = (T_1(\omega), T_2(\omega), \dots, T_N(\omega))^T$  – стохастический  $N$ -вектор стохастических температур в узлах тепловой схемы, то есть элементов в тепловой модели;  $A$  – прямоугольная  $N \times M$ -матрица инцидентий графа тепловой схемы,  $M$  – количество ветвей, матрица инцидентий  $A$  состоит только из элементов 0, +1 и -1;

$G(T, \omega) = \text{diag}(g_1(\omega), g_2(\omega), \dots, g_M(\omega))$  – стохастическая диагональная  $M \times M$ -матрица стохастических проводимостей ветвей  $g_k(\omega)$ ,  $k = 1, 2, \dots, M$ ;  $\Phi(\omega) = (\Phi_1(\omega), \Phi_2(\omega), \dots, \Phi_{N-2}(\omega), 0, 0)^T$  – стохастический

$N$ -вектор стохастических независимых источников тепла (мощностей потребления)  $\Phi_i(\omega)$  активных элементов ЭС;  $T_a(\omega) = (0, 0, \dots, T_e(\omega), T_{a,in}(\omega))^T$  – стохастический  $M$ -вектор известных стохастических температур окружающей среды  $T_e(\omega)$  вне корпуса ЭС и жидкостного потока  $T_{a,in}(\omega)$  на входе в ЭС;  $\Omega$  – пространство элементарных событий  $\omega$  на вероятностном пространстве  $\{\Omega, U, P\}$ ,  $U$  –  $\sigma$ -алгебра подмножеств  $\Omega$ ,  $P$  – вероятность на  $U$ [12];  $(*)^T$  – операция транспонирования.

Матричное уравнение (1) является стохастическим и представляет собой систему стохастических матричных нелинейных алгебраических уравнений, полностью определяющих искомый вектор стохастических стационарных температур  $T(\omega)$  элементов ЭС.

Исчерпывающей характеристикой случайных процессов являются все законы распределения вероятностей всех порядков в любые моменты времени [1, 11, 12], однако, их определение наталкивается на непреодолимые трудности из-за сложности уравнений математической модели (1).

Но для моделирования стохастических тепловых режимов ЭС знание законов распределения стохастических температур элементов ЭС и не требуется, поскольку с приемлемой для практики точностью достаточно знать статистические меры интервально стохастических температур  $T_i(\omega)$ ,  $i = 1, 2, \dots, N$ , элементов ЭС, а именно  $(E\{\cdot\})$  – оператор математического ожидания: математические ожидания  $\bar{T}_i = E\{T_i(\omega)\}$ ; дисперсии  $D_{T_i} = E\{(T_i(\omega) - \bar{T}_i)^2\}$ , где  $T_i(\omega) = T_i(\omega) - \bar{T}_i(t)$  – центрированная стохастическая температура с нулевым математическим ожиданием  $E\{T_i(\omega)\} = 0$  и дисперсией  $D_{T_i}$ ; среднеквадратические отклонения  $\sigma_{T_i} = \sqrt{D_{T_i}}$ ; ковариации  $K_{T_i, T_j} = E\{T_i(\omega) T_j(\omega)\}$  между температурами  $T_i = T_i(\omega)$  и  $T_j = T_j(\omega)$  элементов  $i$  и  $j$  ( $i, j = 1, 2, \dots, N$ ).

По найденным статистическим мерам  $(\bar{T}_i, D_{T_i}, \sigma_{T_i}, K_{T_i, T_j}, i, j = 1, 2, \dots, N)$  далее определяются интервалы  $[T_{Bot, i}, T_{Up, i}]$  охватывающие реальные значения температур элементов ЭС, с нижними  $T_{Bot, i}$  и верхними  $T_{Up, i}$  границами, равными  $T_{Bot, i} = \bar{T}_i - \epsilon \cdot \sigma_{T_i}$ ,  $T_{Up, i} = \bar{T}_i + \epsilon \cdot \sigma_{T_i}$ , где  $\epsilon$  – коэффициент, значение которого с достаточной для инженерной практики точностью оценивается неравенством Чебышева

$$P\left\{T_i(t, \omega) \leq \epsilon \cdot \sigma_{T_i}(t)\right\} \geq 1 - 1/\epsilon^2 \quad [11, 12].$$

## Псевдорешение стохастического уравнения (1)

Рассмотрим стохастическое матричное нелинейное алгебраическое уравнение (1) и умножим обе его части слева на транспонированную матрицу инцидентий  $A^T$ , получим:

$$A^T A G(\omega) A^T \cdot T(\omega) = A^T \cdot \Phi(\omega) + A^T A G(\omega) \cdot T_a(\omega). \quad (2)$$

Матрица  $B = A^T A$  равна произведению двух прямоугольных  $M \times M$ -матрицы  $A^T$  и  $N \times M$ -матрицы  $A$ , причем для тепловых моделей реальных ЭС выполняется соотношение  $N < M$ .

Квадратная матрица  $B$  ранга  $r = \min\{N, M\}$ , представленная в виде  $A^T A$  является ее скелетным разложением [3] и так как  $r < M$  матрица  $B$  является вырожденной и, потому обратной матрицы не имеет.

Вместе с тем, существует ее псевдообратная матрица  $B^+$ , обладающая следующим свойством: для матричной системы уравнений  $Bx = y$  ( $x$  и  $y$  –  $N$ -векторы) с прямоугольной или вырожденной матрицей  $B$  норма невязки  $\|y - Bx\|^2 = \sum_{i=1}^N |y_i - \sum_{j=1}^M b_{ij} x_{ij}|^2$  достигает своего наименьшего среди всех других векторов  $X$  значения на, так называемом, псевдорешении  $X^+ = B^+ Y$ , выражающимся через псевдообратную матрицу  $B^+$ , причем вектор  $X^+$  имеет и наименьшую длину [2, 3].

Псевдообратная матрица  $B^+$  вырожденной матрицы  $B = A^T A$  равна [2, 3]  $B^+ = A^T (A A^T)^{-1} (A A^T)^{-1} A$ , с которой уравнение (2) запишется в виде:

$$G(\omega) A^T \cdot T(\omega) = B^+ A^T (\Phi(\omega) + A G(\omega) \cdot T_a(\omega)). \quad (3)$$

Диагональная матрица  $G(\omega) = \text{diag}\{g_k(\omega)\}_1^M$  имеет для каждой стохастической реализации  $\omega \in \Omega$  обратную диагональную же матрицу  $G^{-1}(\omega) = \text{diag}\{g_k(\omega)^{-1}\}_1^M$ , умножая на которую обе части (3) слева получим для каждого  $\omega \in \Omega$

$$A^T \cdot T(\omega) = G^{-1}(\omega) B^+ A^T (\Phi(\omega) + A G(\omega) \cdot T_a(\omega)). \quad (4)$$

Умножая соотношение (4) слева на матрицу  $A$ , принимая во внимание, что матрица  $C = A A^T$  является квадратной, симметричной и невырожденной (ее ранг  $r = N$  совпадает с ее размерностью) и, следовательно, имеет обратную матрицу  $C^{-1} = (A A^T)^{-1}$ , получим стохастическое псевдорешение  $T^+(\omega)$ :

$$T^+(\omega) = C^{-1} A G^{-1}(\omega) A^T C^{-1} (\Phi(\omega) + A G(\omega) \cdot T_a(\omega)), \quad (5)$$

## Статистические меры

Получим уравнения для определения статистических мер стохастического вектора псевдотемператур  $T^+(\omega)$  элементов ЭС.

Уравнение для вектора математического ожидания температур  $\bar{T}^+$  получается применением оператора математического ожидания  $E\{\cdot\}$ :

$$\bar{T}^+ = C^{-1} A \bar{G}^{-1} A^T C^{-1} (\bar{\Phi} + A \bar{G} \cdot \bar{T}_a), \quad (6)$$

в котором стохастические элементы  $g_k(\omega)$  в  $M \times M$ -матрице  $G(\omega)$ ,  $\Phi_i(\omega)$  в  $N$ -векторе  $\Phi(\omega)$ , температуры как окружающей среды  $T_e(\omega)$  вне корпуса ЭС, так и жидкостного потока  $T_{a,in}(\omega)$  на входе в корпус ЭС в  $M$ -векторе  $T_a(\omega)$ , входящие в уравнение (1) – статистически независимы.

Для определения уравнений для ковариационной матрицы элементов векторов  $T^+(\omega)$ ,  $\Phi(\omega)$  и  $T_a(\omega)$  запишем выражение для транспонированного вектора  $T^+(\omega)$ :

$$(T^+(\omega))^T = (\Phi^T(\omega) + T_a^T(\omega) G(\omega) A^T) C^{-1} A G^{-1}(\omega) A^T C^{-1},$$

с транспонированными векторами  $\Phi^T(\omega)$  и  $T_a^T(\omega)$  для векторов  $\Phi(\omega)$  и  $T_a(\omega)$ , в котором учтено, что матрица  $G(\omega)$  является диагональной, а матрицы  $C = A A^T$  и  $A G(\omega) A^T$  – симметрическими. Уравнение для ковариационной матрицы  $K_{TT}$  определится из общего выражения

$$K_{TT} = E\{T^+(\omega) \cdot (T^+(\omega))^T\} = E\{T^+(\omega) \cdot (T^+(\omega))^T\} - \bar{T}^+ \cdot (\bar{T}^+)^T,$$

относительно централизованного вектора  $T^+(\omega) = T^+(\omega) - \bar{T}^+$  с нулевым математическим ожиданием  $E\{T^+(\omega)\} = 0$ . Тогда уравнение ковариационной матрицы  $K_{TT}$  примет вид:

$$K_{TT} = C^{-1} A \cdot E\{G^{-1}(\omega) A^T C^{-1} (\Phi(\omega) + A G(\omega) \cdot T_a(\omega)) \cdot (\Phi^T(\omega) + T_a^T(\omega) G(\omega) A^T) C^{-1} A G^{-1}(\omega)\} \cdot A^T C^{-1} - \bar{T}^+ \cdot (\bar{T}^+)^T, \quad (7)$$

где вектор  $\bar{T}^+$  определяется выражением (6), а транспонированный ему вектор  $(\bar{T}^+)^T$  – выражением

$$(\bar{T}^+)^T = (\bar{\Phi}^T + \bar{T}_a^T \cdot \bar{G} A^T) C^{-1} A \bar{G}^{-1} A^T C^{-1}.$$

Уравнения относительно статистических мер существенно упрощаются если принять, что температуры окружающей среды  $T_e(\omega)$  и жидкостного потока  $T_{a,in}(\omega)$  на входе в корпус ЭС равны между собой. Такое положение соответствует большинству реальных конструкций ЭС за исключением тех редко используемых конструкций ЭС, внутри корпусов которых нагнетается специально подготовленный теплоноситель с температурой

отличной от температуры окружающей среды. В этом случае, как можно показать, вектор температуры среды можно положить равным нулю, а температуры в узлах тепловой схемы принять равными перепадам узловых температур над температурой окружающей среды  $\Delta T_i(\omega) = T_i(\omega) - T_e(\omega)$ ,  $i = 1, 2, \dots, N$ , при этом распределение потоков тепла по ветвям тепловой схемы не изменится [4, 6], иначе говоря, в уравнениях относительно вектора псевдотемператур  $T^+(\omega)$  будет фигурировать вектор  $\Delta T^+(\omega) = T^+(\omega) - T_a(\omega)$ . Тогда уравнение для ковариационной матрицы  $K_{\Delta T \Delta T} = E\{\Delta T^+(\omega) \cdot (\Delta T^+(\omega))^T\}$  относительно стохастического вектора  $\Delta T^+(\omega)$  может быть получено в виде:

$$K_{\Delta T \Delta T} = C^{-1}A \cdot E\{G^{-1}(\omega)A^T C^{-1}\Phi(\omega) \cdot \Phi^T(\omega)\}C^{-1}AG^{-1}(\omega) \cdot A^T C^{-1} - \Delta \bar{T}^+ \cdot (\Delta \bar{T}^+)^T \quad (8)$$

при этом выражения для векторов  $\Delta \bar{T}^+$  и  $(\Delta \bar{T}^+)^T$  имеют вид:

$$\Delta \bar{T}^+ = C^{-1}A\bar{G}^{-1}A^T C^{-1} \cdot \bar{\Phi}, \quad (\Delta \bar{T}^+)^T = \bar{\Phi}^T \cdot C^{-1}A\bar{G}^{-1}A^T C^{-1}.$$

После определения статистических мер  $\Delta \bar{T}^+$  и  $K_{\Delta T \Delta T}$  вектора перепадов температуры  $\Delta T^+(\omega)$  могут быть определены статистические меры  $\bar{T}^+$  и  $K_{TT}$  искомого вектора  $T^+(\omega)$ , а именно:

$$\bar{T}^+ = \Delta \bar{T}^+ + \bar{T}_a(\omega), \quad K_{\Delta T \Delta T} = K_{TT} - K_{T_a T} - K_{T T_a} + K_{T_a T_a},$$

где  $K_{T_a T} = K_{T_a T}^T$  и  $K_{T_a T_a}$  – ковариационные матрицы равные  $K_{T_a T} = E\{T_a^0(\omega) \cdot (T^+(\omega))^T\}$  и  $K_{T_a T_a} = E\{T_a^0(\omega) \cdot (T_a^0(\omega))^T\}$ .

## Заключение

В работе развит новый метод математического моделирования стохастических распределений температуры в конструкциях ЭС, основанный на понятиях псевдорешения и псевдообратной матрицы. По сравнению с существующими методами, данный подход является универсальным и не ограничен видом функций распределения вероятностей стохастических факторов и параметров элементов ЭС и окружающей среды. Компьютерные вычисления по данному методу не требуют большого количества машинного времени и оперативной памяти. Получены матричные уравнения для определения статистических мер стохастических температур элементов тепловой модели ЭС – вектора математических ожиданий и ковариационной матрицы температур элементов ЭС.

Публикация выполнена по госзаданию ФГУ ФНЦ НИИСИ РАН по теме 0065-2018-0003 «Исследование и разработка моделей и методов математического и компьютерного моделирования тепловых процессов в электронных системах различного назначения в условиях неопределенности» АААА-А18-118032290204-2.

# The stochastic pseudo-solution method for mathematical modeling of thermal modes in electronic systems under uncertainty of determinative parameters

A.G. Madera

**Annotation:** A method of mathematical modeling of stochastic thermal modes in electronic systems, based on the concepts of pseudo-solution and pseudo-inverse matrix, is proposed. In contrast to the existing methods, the developed method is universal and not limited to the type of probability distribution functions of stochastic parameters of the electronic system and the environment. Matrix equations are obtained for the statistical measures of the stochastic temperatures of the elements of the thermal model of the electronic system, namely, the vector of mathematical expectations and the covariance matrix.

**Keywords:** thermal mode, electronic system, pseudo-solution, uncertainty, statistical measures, expectation, variance

## Литература

1. Дж. Адомиан. Стохастические системы. – М.: Мир, 1987.
2. Д.В. Беклемишев. Дополнительные главы линейной алгебры – М.: Лань, 2008.
3. Ф.Р. Гантмахер. Теория матриц – М.: ФИЗМАТЛИТ, 2010.
4. К.С. Демирчян К.С., Л.Р. Нейман., Н.В. Коровкин, В.Л. Чечурин В.Л. Теоретические основы электротехники. Том 1. – СПб: Питер, 2004.
5. С.М. Ермаков, Г.А. Михайлов. Статистическое моделирование. – М.: Наука, 1982.
6. А.Г. Мадера. Моделирование теплообмена в технических системах. – М.: Науч. Фонд «Первая исслед. лаб. им. акад. В.А. Мельникова», 2005.
7. А.Г. Мадера, П.И. Кандалов. Моделирование трехмерных температурных полей в электронных модулях // Программные продукты и системы. 2010. № 2. С. 29 – 33.
8. А.Г. Мадера, П.И. Кандалов. Моделирование температурных полей технических систем в условиях интервальной неопределенности // Тепловые процессы в технике. 2014. Т. 6. № 5. С. 225 – 229.
9. А.Г. Мадера, П.И. Кандалов. Компьютерное моделирование температурных полей технических систем при интервально стохастической неопределенности параметров // Прикладная информатика. 2015. Т. 10. № 1 (55). С. 106 – 113.
10. А.Г. Мадера, П.И. Кандалов. Математическое моделирование интервально стохастических тепловых процессов в технических системах при интервальной неопределенности определяющих параметров // Компьютерные исследования и моделирование. 2016 Т. 8 № 3. С. 501 – 520.
11. В.С. Пугачев. Теория случайных функций. – М.: Наука, 1962.
12. Ю.А. Розанов. Теория вероятностей, случайные процессы и математическая статистика. – М., Наука, 1985.
13. G.N. Ellison. Thermal computations for electronics. Conductive, radiative, and convective air cooling. – NY: CRC Press, 2011.
14. C.J. Keller, V.W. Antonett. Statistical thermal design for computer electronics // Electronic Packaging and Production. 1979. V.19. N.3. P. 55 – 62.
15. A.G. Madera. Interval-Stochastic Thermal Processes in Electronic Systems: Analysis and Modeling //Journal of Engineering Thermophysics. 2017. Vol. 26. N 1. P. 17-28.
16. A.G. Madera. Interval-Stochastic Thermal Processes in Electronic Systems: Analysis and Modeling //Journal of Engineering Thermophysics. 2017. Vol. 26. N. 1. P. 29-38.

# Принципы охлаждения корпусов моноблоков повышенной мощности потребления с применением тепловых труб

А.А. Воробьев<sup>1</sup>, П.И. Кандалов<sup>2</sup>, А.Г. Мадера<sup>3</sup>, О.В. Сердин<sup>4</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

E-mail's: <sup>1</sup> vorobiev@cs.niisi.ras.ru, <sup>2</sup> petrki87@gmail.com, <sup>3</sup> omm@niisi.ras.ru, <sup>4</sup> serdin@cs.niisi.ras.ru

**Аннотация:** Перспективные конструкции моноблоков характеризуются повышенной мощностью потребления  $I_j$  200 Вт. Поэтому традиционные способы охлаждения использующие принудительный конвективный теплообмен в воздух с поверхности радиатора и установленного на нем процессора, не обеспечивают эффективный отвод тепла и уровни температуры кристалла процессора меньших предельно-допустимой. Для высокоэффективного отвода избыточного тепла от процессоров моноблоков с мощностью потребления до 200 Вт необходимо использовать современные технологии охлаждения с применением тепловых труб и, в частности, контурных тепловых труб. В статье рассмотрены результаты моделирования температуры процессора в корпусе моноблока для различных способов охлаждения, включая принудительное воздушное охлаждение, гибкие кондуктивные теплоотводы и контурные тепловые трубы.

**Ключевые слова:** способы охлаждения, корпус моноблока, процессор, контурная тепловая труба, гибкий кондуктивный теплоотвод

Особенность моноблока персонального компьютера состоит в том, что собственно персональный компьютер (ПК) собран в едином блоке, помещенном в монитор компьютера, иными словами, моноблок – это компьютер без системного блока (рис. 1). Компоненты моноблока – материнская плата, интегральные микросхемы (МС) электро-радиоэлементы (ЭРЭ), – те же что и у

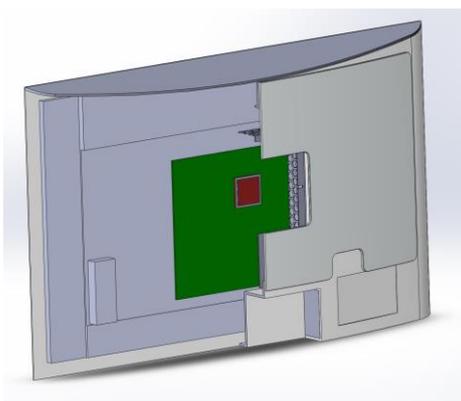


Рис. 1. Конструкция корпуса моноблока. Схематически показаны фрагмент задней стенки корпуса, процессор, плата

обычного ПК, однако, ввиду значительно меньшего конструктивного пространства в корпусе моноблока по сравнению с системным блоком ПК, к системе охлаждения моноблоков предъявляются более жесткие требования как по эффективности охлаждения, так и миниатюризации.

Последнее обуславливается тем, что в замкнутом, предельно ограниченном пространстве моноблока все МС и ЭРЭ, находятся в состоянии значительного теплового взаимодействия между собой. При этом общий температурный фон внутри корпуса моноблока характеризуется существенно более высокими уровнями температуры, чем в обычных ПК с системным блоком. В силу теплового взаимодействия энергопотребляющих компонентов МС и ЭРЭ, между ними возникает эффект воздействия тепловой обратной связи (ТОС). Тепловая обратная связь в МС обуславливается, как известно [1, 2], значительной температурной зависимостью мощности потребления полупроводникового кристалла МС, в результате температура кристалла, возникающая в процессе функционирования МС, приводит к изменению мощности МС, а изменившаяся мощность, в свою очередь, – к изменению температуры МС, замыкая тем самым петлю ТОС. Воздействие ТОС на электрический и тепловой режимы МС и всей электронной системы (ЭС) может быть весьма значительным и пренебрежение ею может приводить к существенным ошибкам при проектировании МС и ЭС. Именно воздействие ТОС, приводящее к взаимовлиянию между температурой и мощностью потребления МС, обуславливает жесткие требования к уровням температур на кристаллах МС и ЭРЭ, которые, чтобы сохранять свою работоспособность и

надежностные характеристики в допустимых пределах, как правило, не должны превышать 125 – 150°C. Причем постоянно наблюдающаяся тенденция роста быстродействия микропроцессоров при одновременном увеличении потребляемой мощности – вплоть до 200 Вт – ставит проблему отведения избыточного тепла от нагреваемых компонентов, установленных в моноблоке, в ряд наиболее актуальных. При этом решение проблемы эффективного отвода избыточной теплоты является зачастую



Рис. 2. Промышленный образец ГКТ  
[<https://thermal-space.com/copper-thermal-straps>]

решающим на этапе разработки и создания конкурентоспособных корпусов моноблоков.

В настоящее время корпус моноблока охлаждается способом принудительного воздушного охлаждения с применением радиатора установленного на мощном процессоре и обдувающего его вентилятора, нагнетающего наружный воздух в корпус моноблока. Численное моделирование температурных распределений внутри корпуса моноблока (разработка НИИСИ РАН) для условий естественной и принудительной конвекции в воздух, при температуре окружающей воздушной среды 20°C и мощности процессора 15 Вт, приведены в табл. 1.

Таблица 1  
Температура процессора в корпусе моноблока при естественном и принудительном

Условия охлаждения процессора в моноблоке	Температура процессора, °C	Средняя температура задней крышки корпуса моноблока, °C
Естественная конвекция	130,7	24,9
Вынужденная конвекция (обдув)	80,5	24
Установка ГКТ	56,8	34,2

конвективном воздушном охлаждении

Для сравнения с данными, приведенными в табл. 1, рассматривался

также вариант способа отвода тепла от процессора с применением гибкого кондуктивного теплоотвода (ГКТ) конструкцией, предложенной в патенте на полезную модель [5] (промышленный образец ГКТ, рис. 2, см. в [<https://copper-thermal-straps>]).

Конструкция ГКТ содержит две теплопроводные пластины (например, из меди) соединенные между собой гибкой теплопроводной связью, представляющей собой жгут, состоящий из большого количества медных тонких проволочек (~ 0,1 мм), обеспечивающих, в совокупности, как высокую степень гибкости соединения с возможностью расположения пластин под любым углом друг относительно друга, так и высокую теплопроводность. Одна из пластин ГКТ крепится к корпусу охлаждаемого процессора в моноблоке, другая – к стоку тепла, – применительно к рассматриваемому моноблоку – к задней стенке моноблока, тепло от которой передается далее путем естественной конвекции окружающему воздуху. Преимуществом ГКТ являются его низкая цена, простота изготовления, устойчивость к вибрациям, нечувствительность к механическим перегрузкам, а также сохранение работоспособности при изменении ориентации в пространстве. К недостаткам ГКТ (в сравнении с тепловыми трубами, в частности с контурными тепловыми трубами (КТТ), рассматриваемыми ниже) могут быть отнесены более низкие значения отводимой мощности и, как следствие, более высокие уровни температуры на процессоре, по сравнению с КТТ. Так, согласно результатам



Рис. 3. Принципиальная схема КТТ [3]

проведенного моделирования, температура процессора с мощностью тепловыделения 15 Вт, при температуре окружающей среды 20°C, и конструкции ГКТ с длиной гибкой теплопроводной связи 40 мм и тепловым сопротивлением 0,82 К/Вт, температура процессора составляет 56,8°C. В то время как применение КТТ позволяет снизить температуру процессора до 42°C. (см. ниже).

Отметим также, что тепловое сопротивление ГКТ линейно растет с увеличением длины гибкой теплопроводной связи, однако, может быть существенно снижено посредством увеличения числа медных проволочек в жгуте гибкой теплопроводной связи.

относится применение низкотемпературных тепловых труб и, в частности, контурных тепловых труб (КТТ), принципиальная схема которой приведена на рис. 3 [3], а устройство и принцип действия подробно рассмотрены в литературе [4]. К главным преимуществам



Рис. 4. Моноблок охлаждаемый КТТ: а – внешний вид, б – вид со стороны задней крышки, в – КТТ [3]

Из табл. 1 следует также, что температура процессора в моноблоке в условиях естественной конвекции ( $130,7^{\circ}\text{C}$ ) уже при температуре внешней среды  $20^{\circ}\text{C}$  превышает допустимое значение, а при обдуве вентилятором и температуре среды  $20^{\circ}\text{C}$  составляет  $80,5^{\circ}\text{C}$ . При температуре воздуха снаружи моноблока свыше  $40^{\circ}\text{C}$  температура процессора будет превышать  $120^{\circ}\text{C}$ , что также недопустимо, поскольку не оставляет достаточного запаса до предельно-допустимой температуры кристалла процессора. Применение ГКТ, в свою очередь, позволяет эффективно передавать избыточное тепло к стоку в пределах присущих ему возможностей как по длине гибкой теплопроводной связи, так и величине теплового сопротивления. Необходимо отметить, что использование в качестве основного способа снижения температуры в корпусе моноблока принудительного воздушного охлаждения с применением вентиляторов, является неперспективным. Это обуславливается как тенденцией к росту мощности потребления процессоров и неспособностью данного способа охлаждения отводить значительные потребляемые мощности при повышенных температурах среды, так и низкой надежностью и значительными шумовыми воздействиями, которыми сопровождается работа вентилятора.

Таким образом, для эффективного отвода тепла от процессоров со значительной мощностью потребления, устанавливаемых в моноблоках, необходимо использовать новые более эффективные способы охлаждения. К таким перспективным способам охлаждения, получающим все большее распространение,

КТТ, по сравнению с другими способами охлаждения, в том числе и иными конструкциями тепловых труб, относятся:

- низкое тепловое сопротивление (менее  $0,25$  К/Вт),
- высокие значения отводимой мощности (свыше  $300$  Вт) при низких значениях температуры испарителя (не выше  $90^{\circ}\text{C}$ ). Как указывается в [3] при мощности потребления процессора  $140$  Вт (CPU Intel Core i7-5820K) температура в зоне источника тепла составляет всего  $45^{\circ}\text{C}$ , что

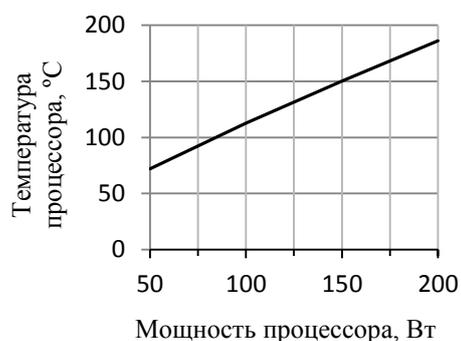


Рис. 5. Температура процессора в моноблоке в зависимости от его мощности

- существенно ниже допустимой температуры,
- осуществление концепции удаленного стока тепла, когда расстояние между источником тепла и его стоком в систему охлаждения, может составлять несколько метров,
- малые габариты КТТ, отсутствие шумов, высокая надежность.

Исследования и разработка конструкций КТТ для охлаждения ЭС различного назначения и моноблока (рис. 4) со значительным энергопотреблением

процессора (разработка НИИСИ РАН), проводились в лаборатории теплопередающих устройств Института теплофизики УрО РАН.

В конструкции моноблока, рассматриваемой в данной статье, тепло, выделяемое процессором, поступало к установленному на корпусе процессора медному интерфейсу, а от него – к КТТ и конденсаторов, являющийся частью конструкции КТТ. Охлаждение конденсаторов КТТ осуществлялось посредством естественной конвекции с задней алюминиевой стенки моноблока в окружающую среду, при этом трубка конденсаторов была припаяна к пазам, выполненным на задней стенке корпуса (рис. 4, в). Ввиду того, что в рассматриваемом реальном корпусе моноблока мощность потребления процессора составляла 15 Вт, поверхности задней алюминиевой стенки, служившей радиатором, было достаточно, чтобы обеспечить необходимое рассеивание тепла в окружающую среду путем естественной конвекции. Результаты эксперимента показали, что температура процессора в моноблоке не превышает 42°C.

Полученная в результате моделирования (Solid-Works) зависимость температуры процессора от его мощности для теплового сопротивления КТТ равного  $R_{КТТ} = 0,2 \text{ К/Вт}$  и температуре окружающей среды 20°C, приведена на рис. 5. Из приведенной зависимости следует, что температура кристалла процессора не будет превышать предельно-допустимую температуру (150°C) вплоть до мощности процессора 150 Вт.

Для того чтобы температура процессора не превышала предельно-допустимую для мощностей порядка 200 Вт необходимо снижать тепловое сопротивление КТТ и использовать более эффективные радиаторы для конвективного отвода тепла от задней стенки корпуса моноблока.

Публикация выполнена по госзаданию ФГУ ФНЦ НИИСИ РАН по теме 0065-2018-0018 «Разработка общих принципов и методик создания корпусов моноблоков и серверов в соответствии с требованиями по уровням электромагнитных излучений и отводу тепла» (Per № АААА-А18-118041190173-4).

## Principles of cooling of monoblock enclosure with increased consumed power by heat pipes

A.A. Vorobyev, P.I. Kandalov, A.G. Madera, O.V. Serdin

**Abstract:** Perspective monoblock enclosure designs are characterized by an increased power consumption of up to 200 watts. Therefore, traditional methods of cooling using forced convective heat transfer into the air from the surface of the radiator and the processor installed on it, do not provide effective heat exchange and the temperature levels of the processor crystal lower than the maximum allowable. For highly efficient removal of excess heat from monoblock processors with power consumption up to 200 watts, it is necessary to use modern cooling technologies using heat pipes and, in particular, loop heat pipes. The article describes the results of modeling the temperature of the processor in the monoblock enclosure for various cooling methods, including forced air cooling, flexible conductive heat sinks and loop heat pipes.

**Keywords:** cooling methods, monoblock enclosure, processor, loop heat pipe, flexible conductive heat sink

### Литература

1. А.Г. Мадера. Моделирование воздействия тепловой обратной связи на тепловые процессы в электронных системах // Компьютерные исследования и моделирование. 2018. Т. 10. №4. С. 483 – 494
2. А.Г. Мадера. Тепловые процессы в электронных системах в условиях влияния эффекта тепловой обратной связи // Тепловые процессы в технике. 2018. Т. 10. № 3-4. С. 144 – 151
3. Ю.Ф. Майданик. Контурные тепловые трубы – высокоэффективные теплопередающие устройства для систем охлаждения электроники // Электроника: Наука, технология, бизнес. 2017. № 6 (166). С. 122 – 130
4. Ю.Ф. Майданик. Разработка и экспериментальные исследования миниатюрной контурной тепловой трубы с тонким плоским испарителем // Тепловые процессы в технике. 2017. №8. С. 346 – 352
5. Устройство охлаждения и отвода тепла от компонентов электронных систем. Патент на полезную модель № 117056 // А.Г. Мадера, В.И. Гречишников, М.Ю. Гнитиев.

# Исследование влияния параметров клинового механизма на обеспечение теплоотвода модуля 6U в магистрально-модульной системе типа VPX

А.В. Зуйков<sup>1</sup>, А.А. Воробьев<sup>2</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия, E-mail's: <sup>1</sup> [zuykov@cs.niisi.ras.ru](mailto:zuykov@cs.niisi.ras.ru), <sup>2</sup> [vorobiev@cs.niisi.ras.ru](mailto:vorobiev@cs.niisi.ras.ru)

**Аннотация:** В данной статье рассматриваются подходы к обеспечению теплопередачи через клиновой механизм модуля 6U на корпус магистрально-модульной системы типа VPX с кондуктивным охлаждением. Рассмотрены варианты улучшения теплопередачи без изменения конструкции клинового механизма.

**Ключевые слова:** клиновой механизм, системы VPX, модуль 6U, тепловое сопротивление, перенос тепла.

## Введение

Магистрально-модульные архитектуры на базе стандартов VPX являются преемниками зарекомендовавшей себя технологии VME. Недостатком технологии VME является ее несоответствие современным требованиям по внутрисистемной пропускной способности. VPX-системы построены на базе высокоскоростных интерфейсов и преимущественно используются в оборонной и аэрокосмической отраслях. Наиболее остро проблема обеспечения теплоотвода встает в VPX-системах с кондуктивным охлаждением, для которых, потребление модулей 6U может достигать более 50 Вт (стандартом максимальное потребление ограничено значением 100 Вт).

## Клиновой механизм

VPX-система, представленная на рисунке 1, состоит из 3-х процессорных модулей, с потреблением 36 Вт каждый (см. рисунок 2).



Рисунок 1. VPX-система с кондуктивным охлаждением



Рисунок 2. Процессорный модуль (2 процессора, общая мощность 36 Вт)

С основных энергопотребителей (микросхемы, преобразователи питания и пр.), расположенных на печатной плате процессорного модуля, выделяемое тепло снимается при помощи радиатора, установленного на плату. Для обеспечения теплоотвода с радиатора модуля на корпус изделия применяется клиновой механизм (см. рисунок 3), который так же предназначен для фиксации модуля в корпусе ЭВМ.

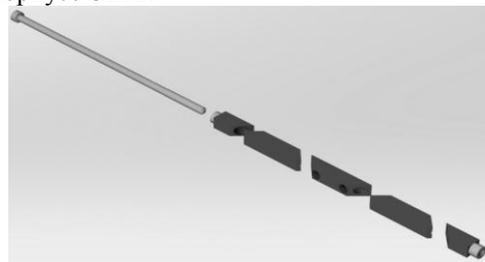


Рисунок 3. Клиновой механизм

Типовой клиновой механизм состоит из 5 секций, стягивание которых между собой осуществляется при помощи винта. При расширении, клиновой механизм

надежно фиксирует модуль в корпусе ЭВМ и обеспечивает теплопередачу. Из рисунка 4 видно, что часть тепла передается от модуля

на корпус через непосредственный контакт радиатора с корпусом, другая часть через клиновый механизм.

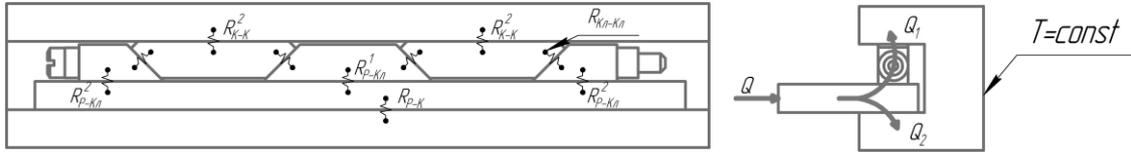


Рисунок 4. Тепловые сопротивления в механизме фиксации модуля

Качество передачи тепла с модуля на корпус ЭВМ зависит от конструкции клинового механизма, силы зажима, материала и покрытия клинового механизма.

Для определения возможности улучшения теплопередачи между вычислительным модулем и корпусом ЭВМ было проведено исследование, в котором анализировалось изменение значений тепловых сопротивлений между модулем и корпусом ЭВМ, при изменении таких характеристик, как материал деталей клинового механизма, покрытие клинового механизма и усилие затяжки винта.

Для определения тепловых сопротивлений в контактах деталей клинового механизма, критическое значение имеют силы, воздействующие по нормали к контактирующим поверхностям. Для их определения воспользуемся схемой распределения сил, приведенной на рисунке 7.

Для определения наиболее интересующих нас сил  $N$ ,  $W$ ,  $F_{сж}$  составим уравнения баланса сил (1):

$$\begin{cases} F_{сж} = P + F_{мп1} \\ W = N_1 \end{cases} \quad (1)$$

$$\begin{cases} P = W \cdot \operatorname{tg}(\alpha + \varphi) \\ F_{мп1} = N_1 f = W \cdot \operatorname{tg} \varphi \end{cases} \quad (2)$$

где,  $F_{сж} = M / dk$  – усилие от затяжки винта,  $M$  – момент затяжки,  $d$  – диаметр тела винта,  $k$  – коэффициент трения,  $P$  – сила выталкивающая клин,  $F_{мп1}$  – сила трения у основания клина,  $W$  – подъемная сила со стороны клина,  $f = \operatorname{tg} \varphi$  – коэффициент трения деталей клинового механизма, выражаемый через угол трения.

$$W = F_{сж} \frac{1}{\operatorname{tg}(\alpha + \varphi) + \operatorname{tg} \varphi} \quad (3)$$

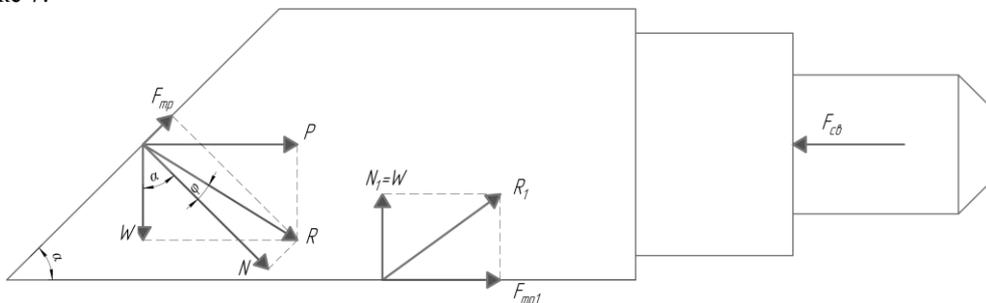


Рисунок 5. Силы, действующие в клиновом механизме.

Следовательно, в соответствии с рисунком 5 нормальную силу вычислим, как (4):

$$\begin{cases} R = \frac{W}{\cos(\alpha + \varphi)} \\ N = R \cos \varphi \end{cases} \quad (4)$$

Расчетные данные для разных конфигураций клинового механизма приведены в таблице 1.

В соответствии с [2] усилия затяжки винта выбирались исходя из имеющейся конструкции клинового механизма, а также из предельных возможностей для винта данного размера. Выбор покрытий клинового механизма был обусловлен их широкой распространенностью и

механическими (защитными) свойствами. В исходной конфигурации клинового механизма используется анодное окисление (Ан.Окс), а в качестве его альтернативы было выбрано никелевое покрытие, полученное химическим способом (Хим.Никель), обладающее повышенными твердостью и теплопроводностью по сравнению с Ан.Окс.

Из-за различия коэффициентов трения покрытий, при одинаковом моменте затяжки винта нормальное усилие  $N$  между клиньями, покрытыми Хим.Никель, возросло на 13% по сравнению с клиньями, покрытыми Ан.Окс, а общая прижимная сила возросла на 37%.

Таблица 1. Воздействующие усилия для различных конфигураций клинового механизма.

№ расчета	$T_{кр}$ , Нм	$F_{сж}$ , кН	$N$ , кН	$F_{св}$ , кН	Покрытие	Материал клинов
1	5,4	9,89	5,03	5,4	Ан.Окс	Д16Т
2	5,4	13,6	5,71	5,4	Хим.Никель	Д16Т
3	3,2	5,85	2,89	3,2	Ан.Окс	Д16Т
4	3,2	8,05	3,38	3,2	Хим.Никель	Д16Т
5	5,4	9,89	5,03	5,4	Ан.Окс	АД31Т1
6	5,4	13,6	5,71	5,4	Хим.Никель	АД31Т1
7	3,2	5,85	2,89	3,2	Ан.Окс	АД31Т1
8	3,2	8,05	3,38	3,2	Хим.Никель	АД31Т1

где,  $T_{кр}$  – крутящий момент затяжки винта,  $F_{сж}$  – сила расширения клинового механизма,  $N$  – нормальная сила между сегментами клинового механизма,  $F_{св}$  – усилие, развиваемое винтом.

Определение тепловых сопротивлений в контактах деталей проводилось в соответствии с методикой [1].

Тепловая проводимость контакта  $\alpha_k$  может быть представлена, как сумма проводимости фактического контакта  $\alpha_m$  и проводимости межконтактной среды  $\alpha_c$ , тогда проводимость контакта:  $\alpha_k = \alpha_m + \alpha_c$ , или выражая через тепловые сопротивления:

$$\frac{1}{R_k} = \frac{1}{R_m} + \frac{1}{R_c}. \text{ Процесс расчета контактной проводимости подробно описан в пособии [1].}$$

Полученные данные были использованы для расчета тепловых сопротивлений в зонах контактов деталей клинового механизма, значения которых приведены в таблице 2:

Таблица 2. Тепловые сопротивления контактов для различных конфигураций клинового механизма.

№ расчета	$R_{P-K}$ , м <sup>2</sup> К/Вт	$R_{P-Kl}^1$ , м <sup>2</sup> К/Вт	$R_{P-Kl}^2$ , м <sup>2</sup> К/Вт	$R_{Kl-Kl}$ , м <sup>2</sup> К/Вт	$R_{K-K}^2$ , м <sup>2</sup> К/Вт
1	$1,56 \cdot 10^{-4}$	$7,60 \cdot 10^{-5}$	$1,16 \cdot 10^{-4}$	$4,70 \cdot 10^{-5}$	$1,30 \cdot 10^{-4}$
2	$1,38 \cdot 10^{-4}$	$7 \cdot 10^{-5}$	$1,03 \cdot 10^{-4}$	$4,2 \cdot 10^{-5}$	$1 \cdot 10^{-4}$
3	$1,91 \cdot 10^{-4}$	$8,2 \cdot 10^{-5}$	$1,4 \cdot 10^{-4}$	$6,55 \cdot 10^{-5}$	$1,37 \cdot 10^{-4}$
4	$1,68 \cdot 10^{-4}$	$7,78 \cdot 10^{-5}$	$1,25 \cdot 10^{-4}$	$6 \cdot 10^{-5}$	$1,21 \cdot 10^{-4}$
5	$1,56 \cdot 10^{-4}$	$7,44 \cdot 10^{-5}$	$1,12 \cdot 10^{-4}$	$4,56 \cdot 10^{-5}$	$1,09 \cdot 10^{-4}$
6	$1,38 \cdot 10^{-4}$	$6,9 \cdot 10^{-5}$	$1 \cdot 10^{-4}$	$4,03 \cdot 10^{-5}$	$9,73 \cdot 10^{-4}$
7	$1,91 \cdot 10^{-4}$	$8,01 \cdot 10^{-5}$	$1,34 \cdot 10^{-4}$	$6,28 \cdot 10^{-5}$	$1,31 \cdot 10^{-4}$
8	$1,69 \cdot 10^{-4}$	$7,6 \cdot 10^{-5}$	$1,2 \cdot 10^{-4}$	$5,78 \cdot 10^{-5}$	$1,17 \cdot 10^{-4}$

где,  $R_{P-K}$  – тепловое сопротивление в контакте радиатора с корпусом,  $R_{P-Kl}^1$  – тепловое сопротивление в контакте центрального неподвижного клина с радиатором,  $R_{P-Kl}^2$  – тепловое сопротивление в контакте боковых клиньев с радиатором,  $R_{Kl-Kl}$  – тепловое сопротивление между клиньями,  $R_{K-K}^2$  – тепловое сопротивление в контакте клиньев с корпусом

Для определения эффективности теплопередачи в клиновом механизме и контакте радиатора с корпусом был проведен расчет с использованием САПР SolidWorks. В расчете были использованы ранее полученные данные, приведенные в таблице 2. Учет теплового сопротивления, создаваемого покрытиями, был произведен в расчетной среде SolidWorks, в расчете контактных сопротивлений по методике [1] учет внешних покрытий деталей велся с помощью задания их механических характеристик. Геометрическое представление модели, используемой для расчета приведено на рисунке 4. Расчет проводился

без учета теплопередачи в окружающую среду при следующих условиях, сопоставимыми с реальными условиями эксплуатации модулей:

- Мощность тепловыделения со стороны модуля  $Q = 18$  Вт;
- Температура стенки корпуса  $T = 60$  °С;
- Материал радиатора модуля: Д16Т;
- Покрытие радиатора модуля: Ан.Окс.;
- Материал корпуса: Д16Т;
- Покрытие корпуса: Ан.Окс.

Примеры эпюр распределения температур, полученных при расчете клинового механизма исходной конфигурации, приведены на рисунках 5, 6.

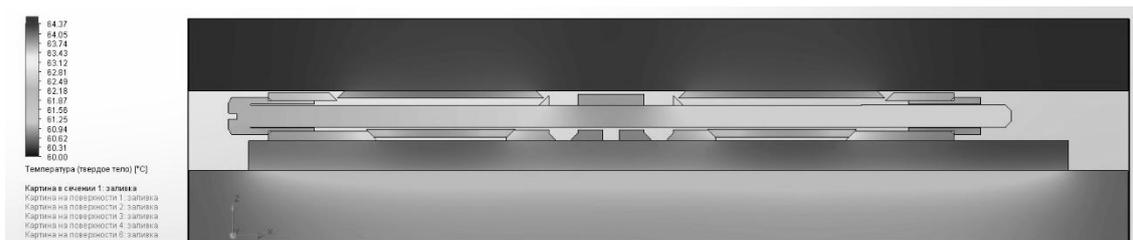


Рисунок 6. Эюра распределения температур в сечении модуля с клиновым механизмом (исходные параметры клинового механизма)



Рисунок 7. Эюра распределения температур на поверхности модуля с клиновым механизмом (исходные параметры клинового механизма)

При проведении расчета были получены параметры, характеризующие общее тепловое сопротивление как клинового механизма, так и контакта радиатора с корпусом. Оцениваемыми факторами являлись:  $R_{Кл\Sigma}$  - суммарное тепловое сопротивление клинового механизма, которое вычислялось по формуле  $R_{Кл\Sigma} = \frac{\Delta T_1}{Q_1}$ , где  $\Delta T_1$  - разница температур между радиатором модуля и корпусом в зонах контакта с клиновым механизмом,  $Q_1$  - тепловой поток, через площадь контактирования клинового механизма с

радиатором модуля.  $R_{P-К\Sigma} = \frac{\Delta T_2}{Q_2}$  - где  $\Delta T_2$  - разница температур между радиатором модуля и корпусом в зоне их контакта,  $Q_2$  - тепловой поток, через площадь контактирования корпуса с радиатором модуля. Для различных вариантов исполнения клиновых механизмов была произведена оценка изменения тепловых сопротивлений, для клинового механизма -  $\Delta R_{Кл\Sigma}$ , для радиатора -  $\Delta R_{P-К\Sigma}$ . Все перечисленные параметры приведены в таблице 3.

Таблица 3. Оценка результатов вычислений.

№ расчета	$R_{Кл\Sigma}$ , К/Вт	$R_{P-К\Sigma}$ , К/Вт	$Q_1$ , Вт	$Q_2$ , Вт	$\Delta R_{Кл\Sigma}$ , %	$\Delta R_{P-К\Sigma}$ , %
1	4.12	0.21	1.95	16.04	3.6	13.28
2	3.03	0.19	2.51	15.48	29.0	20.12
3	4.27	0.24	1.99	16	0	0
4	3.28	0.22	2.43	15.56	23.22	8.73
5	3.71	0.21	2.14	15.85	13.07	13.28
6	2.67	0.19	2.76	15.23	37.35	21.63
7	3.88	0.24	2.15	15.84	9	0
8	2.90	0.22	2.69	15.30	32.1	8.35
9	4.12	0.21	1.95	16.047	3.6	13.28

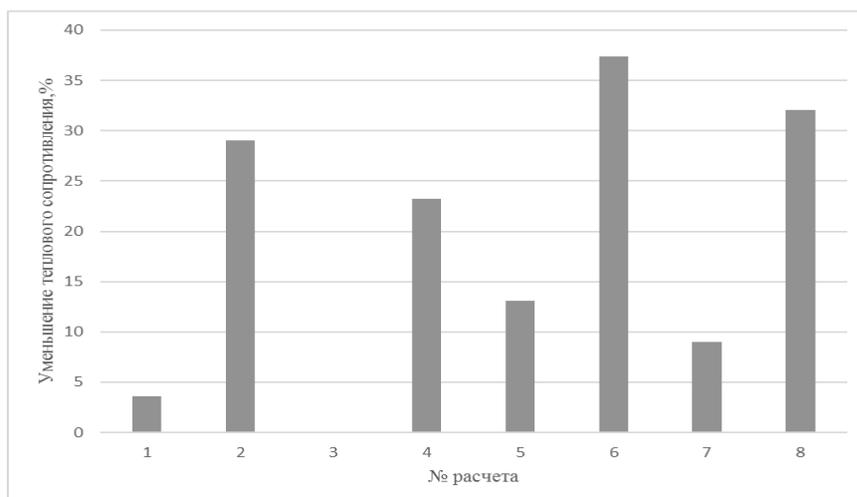


Рисунок 8. Уменьшение теплового сопротивления  $R_{Kt\Sigma}$  в процентном соотношении

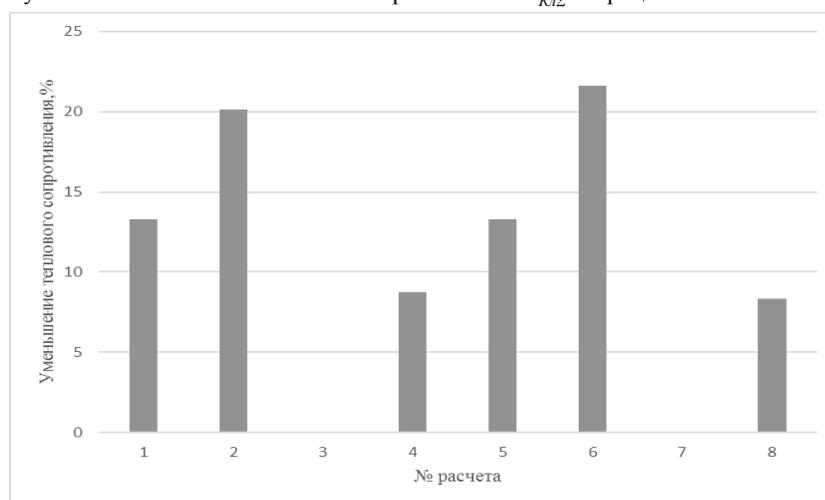


Рисунок 9. Уменьшение теплового сопротивления  $R_{P-K\Sigma}$  в процентном соотношении

## Вывод:

В ходе проведенной работы были проанализированы возможности по улучшению тепловых параметров клинового механизма без замены конструкции. По графикам, представленным на рисунках 8-9 можно оценить вклад каждого изменения в уменьшение теплового сопротивления между вычислительным модулем и корпусом. Так путем анализа данных таблицы 3 можно заключить, что увеличение крутящего момента затяжки винта позволило уменьшить тепловые сопротивления клинового механизма и корпуса соответственно на 3,6% и 13,28%, замена покрытия Ан. Окс на более теплопроводное, но в то же время и твердое Хим.Никель уменьшило значения тепловых сопротивлений на 23,2% и 8,7%. Замена

материала клиньев позволила уменьшить суммарное тепловое сопротивление клинового механизма на 9%. Комбинирование этих преимуществ позволит сильнее уменьшать тепловое сопротивление перехода радиатор-корпус, что позволит повысить надежность и сбоеустойчивость вычислительных машин с кондуктивной системой охлаждения.

Публикация выполнена в рамках государственного задания ФГУ ФНЦ НИИСИ РАН (проведение фундаментальных научных исследований) по теме № 0065-2018-0018 «Разработка общих принципов и методик создания корпусов моноблоков и серверов в соответствии с требованиями по уровням электромагнитных излучений и отводу тепла» (АААА-А18-118041190173-4).

# Providing heat dissipation with 6U modules to the case of the VPX-type modular system

A.V. Zuikov, A.A. Vorobyev

**Abstract:** This article discusses approaches to providing heat transfer through the wedge mechanism of the 6U module to the case of a VPX-type modular system with conductive cooling. The options for improving heat transfer without changing the design of the wedge mechanism are considered.

**Keywords:** wedge mechanism, wedge lok, VPX systems, module 6U, heat transfer, thermal resistance.

## Литература

1. В.М.Попов. Теплообмен в зоне контакта разъемных и неразъемных соединений. М., «Энергия», 1971.
2. Затяжка резьбовых соединений. Нормы затяжки и технические требования: РД 37.001.131-89.

# Обзор новых интерпретационных моделей при гидродинамических исследованиях скважин

С.Г. Вольпин<sup>1</sup>, А.Г. Дяченко<sup>2</sup>, Н.П. Ефимова<sup>3</sup>,

О.В. Ломакина<sup>4</sup>, Ю.Б. Чен-лен-сон<sup>5</sup>

ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

E-mail's: <sup>1</sup>[sergvolpin@gmail.com](mailto:sergvolpin@gmail.com), <sup>2</sup>[adyachenko@bk.ru](mailto:adyachenko@bk.ru), <sup>3</sup>[efinatka@gmail.com](mailto:efinatka@gmail.com),

<sup>4</sup>[ovllom@mail.ru](mailto:ovllom@mail.ru), <sup>5</sup>[chen-len-son@mail.ru](mailto:chen-len-son@mail.ru)

**Аннотация:** Настоящая статья посвящена обзору новых интерпретационных моделей при гидродинамических исследованиях скважин, среди которых наибольший интерес представляют модели горизонтальной скважины с многостадийным гидроразрывом пласта (МГРП) и скважины сложной конфигурации. Для модели горизонтальной скважины с МГРП приведен диагностический график, примеры расчетов, а также формулы для расчета изменения дебита для анизотропного пласта. Приведены формулы и пример расчета для модели скважины сложной конфигурации и пласта малой толщины, а также алгоритм расчета установившейся и неустановившейся фильтрации для любых типов скважин с любыми работающими участками.

**Ключевые слова:** гидродинамические исследования скважин (ГДИС), гидроразрыв пласта (ГРП), горизонтальная скважина, скважины сложной конфигурации.

## Введение

Гидродинамические исследования скважин (ГДИС) являются одним из основных способов изучения фильтрационно-емкостных свойств пласта, его геометрии, свойств призабойной зоны пласта, контроля разработки месторождений. При проведении ГДИС регистрируется забойное давление в скважине и ее дебит при различных режимах работы скважины (включая полную остановку и пуск в работу). Затем кривые забойного давления и дебита совместно анализируются с учетом законов фильтрации флюидов в пористой среде с целью определения параметров пласта.

Законы фильтрации жидкости и газа в пористой среде базируются на основных положениях механики сплошных сред, которая рассматривает коллектор в толще горной породы как сплошную деформируемую среду. Основой математической модели фильтрации является уравнение пьезопроводности (уравнение диффузии). Вывод его основан на трех фундаментальных законах - сохранения (массы, импульса) и уравнении состояния при определенных допущениях (изотропность коллектора, однофазность потока и малая сжимаемость флюида, проницаемость коллектора и вязкость флюида не зависят от давления, влияние силы тяжести игнорируется).

Если наложить определенные начальные, внутренние (на скважине) и внешние граничные условия, то можно решить уравнение и получить аналитическую модель, характеризующую перераспределение давления в пласте.

За последние 50 лет математиками получены и опубликованы аналитические решения для самых разнообразных моделей. Для интерпретации результатов ГДИС применяют значительное их число. Большинство моделей являются комплексными и включает в себя элементы из следующих классов: скважина, пласт, граница, табл.1 [1].

Таблица 1  
Интерпретационные модели для ГДИС

Модели скважины	Модели пласта	Модели границ (непроницаемых и постоянного давления)
1. Вертикальная 2. Наклонная 3. С частичным вскрытием 4. Горизонтальная 5. Вертикальная с ГРП 6. Горизонтальная с ГРП	1. Одно-родный 2. Двойная пористость 3. Двойная проницаемость 4. Радиальный композитный 5. Линейный композитный	1. Бесконечный пласт 2. Одна граница 3. Две параллельные границы 4. Две пересекающиеся границы 5. Круговая граница 6. Прямоугольная граница

Диагностика модели - важнейший шаг в процессе анализа и интерпретации данных ГДИС, когда сложные траектории движения жидкости в пласте могут схематизироваться простейшими одномерными фильтрационными потоками или их различными комбинациями. Простейшими одномерными фильтрационными потоками являются: плоскорадиальный, прямолинейно-параллельный и билинейный фильтрационные потоки, рис. 1, [1].

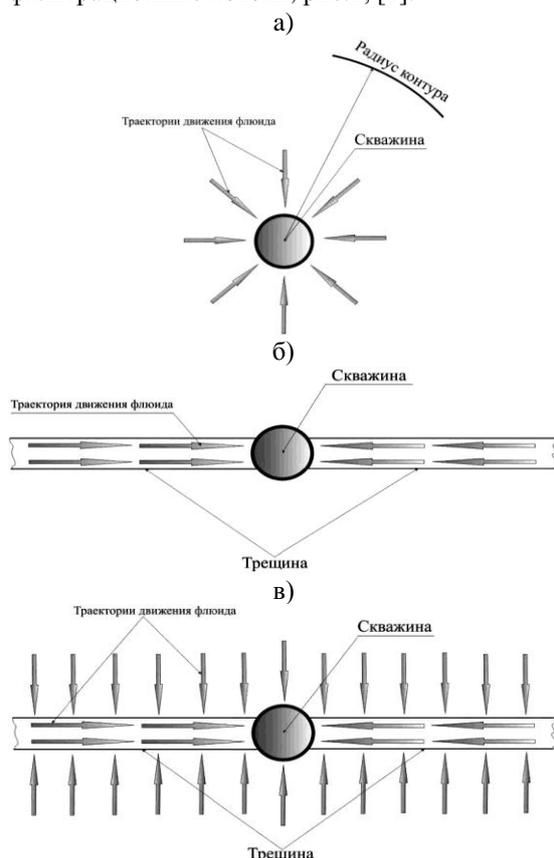


Рис. 1. Простейшие одномерные фильтрационные потоки [1]

- а) плоскорадиальный фильтрационный поток  
 б) прямолинейно-параллельный (линейный) фильтрационный поток  
 в) билинейный фильтрационный поток

Диагностика основана на выявлении режимов течения, которые имеют характерные признаки на бипологарифмическом графике производной Бурде (Bourdet) [2]: прямолинейные участки с различными углами наклона.

На основе выявленных режимов течения можно судить о модели изучаемого пласта.

Когда сложную задачу невозможно представить простым лаконичным решением, ее надо аппроксимировать путем разделения решения на этапы: либо временные шаги, либо кусочки - граница / сегмент скважины / сегмент разлома. Тогда решение предстанет в виде интегральной функции или повременного обра-

щения матрицы. Такое решение называется полуаналитическим, поскольку некоторые компоненты определяются аналитическим способом, однако собираются вместе численным.

Анализ современных научно-технических публикаций по нефтегазовым технологиям (SPE, JPSE, «Нефтяное Хозяйство» и др.) свидетельствует об интересе к модели горизонтальной скважины с многостадийным гидроразрывом пласта (МГРП) и скважины сложной конфигурации.

Доказано, что горизонтальная скважина с МГРП является одной из ключевых технологий успешной разработки плотных (низкопроницаемых) пластов.

В 1984 году была представлена первая математическая модель для анализа продуктивности горизонтальных скважин, пересекающих трещины, в которой были определены течения в матрице породы и трещине, а затем объединены оба течения для получения уравнения радиальной фильтрации для всего пути потока от внешней границы до ствола скважины [3]. С тех пор было немало публикаций, часть из них приведена в работе [4].

Различными исследователями предложено несколько аналитических моделей для прогнозирования дебита горизонтальных скважин с многостадийным гидроразрывом [5-7], что позволяет интерпретировать длительные записи забойного давления и дебита, регистрируемые при работе таких скважин.

Исследование горизонтальных скважин с МГРП методом восстановления давления (регистрация КВД – кривой восстановления давления) не представляется возможным из-за того, что в низкопроницаемых коллекторах такая скважина будет выходить на радиальный режим притока месяцы, а возможно – годы. Остановка скважины на такое время неприемлема. Поэтому анализ работы таких скважин становится единственным доступным методом ГДИС.

## 1. Модель горизонтальной скважины с многостадийным гидроразрывом пласта

Приток к горизонтальной скважине с МГРП характеризуется несколькими последовательно сменяющимися друг друга режимами фильтрации, рис. 2, отличающимися типом симметрии линий тока: линейный, первый переходный (эллиптический), ранний радиальный, второй переходный, псевдордиальный. При диагностике всех режимов течения определяют полный спектр параметров пласта [8].

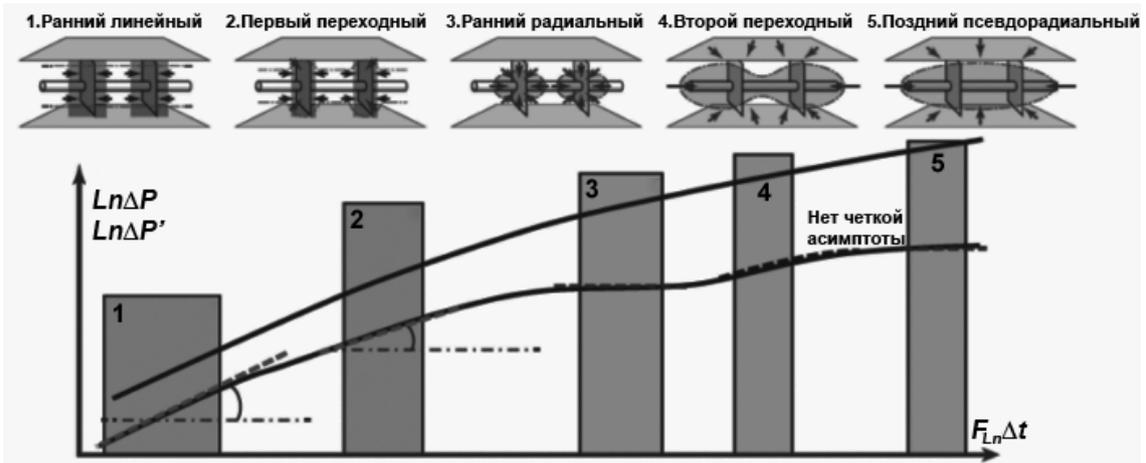


Рис. 2. Log-Log диагностика характерных режимов фильтрации для горизонтальной скважины с МГРП [8]

В работе [9] был разработан алгоритм по определению количества работающих трещин ГРП, который включает в себя сопоставление проницаемостей (гидропроводностей), полу-

ченных по раннему и позднему радиальному режиму фильтрации. Примеры расчетов, полученных по авторской методике [8, 9] представлены на рис. 3 и рис. 4.

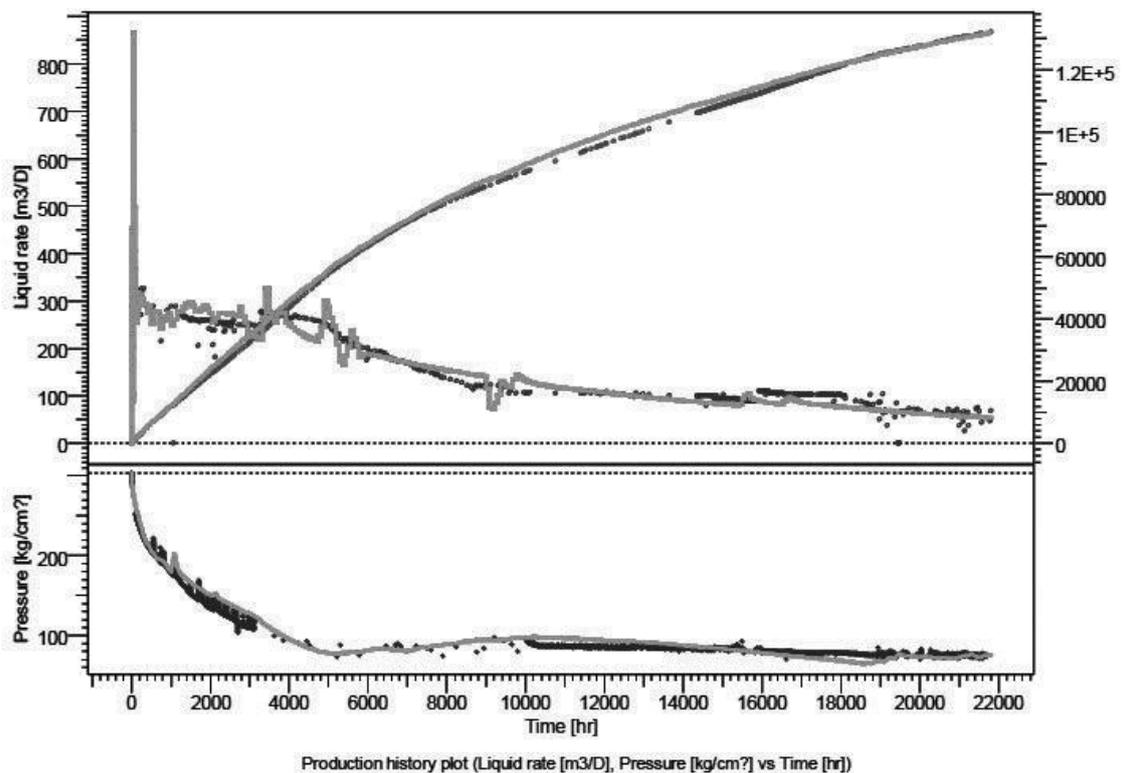


Рис. 3. Совмещение фактических и расчётных параметров дебита, накопленной добычи и давления по скв. \*88G [9]

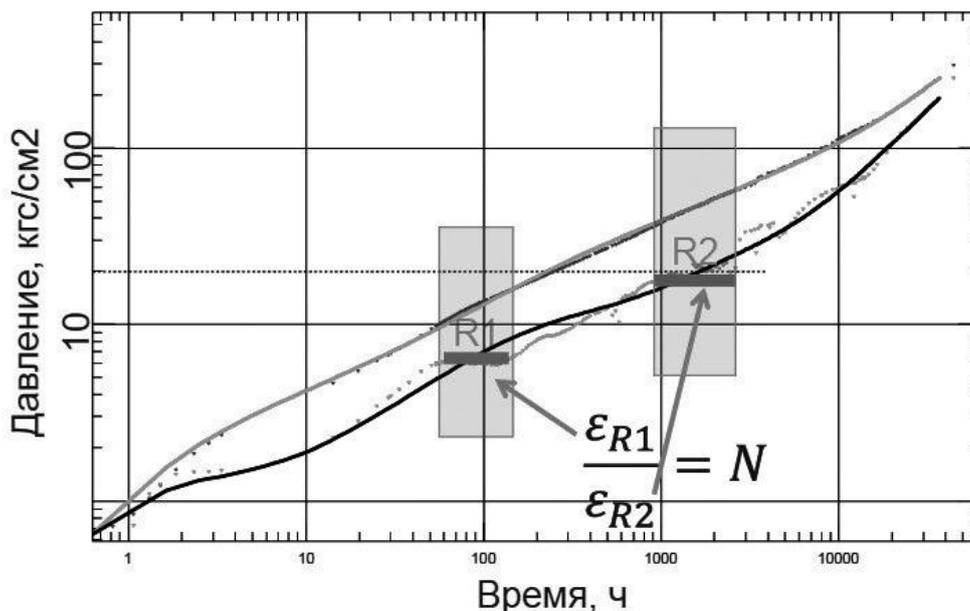


Рис. 4. Совмещение фактических и расчётных параметров приращения давления и производной давления по скв. \*88G в билогарифмических координатах [9]:

$\varepsilon_{R1}$  - гидропроводность раннего радиального режима фильтрации  
 $\varepsilon_{R2}$  - гидропроводность позднего радиального режима фильтрации

В работе [10] представлена модель динамики притока жидкости к трещинам гидравлического разрыва пласта, перпендикулярным к стволу скважины, для анизотропного пласта и предложена формула для расчета изменения дебита горизонтальной скважины с МГРП от времени.

В модели не учитывается влияние схемы размещения нагнетательных скважин относительно добывающей горизонтальной с МГРП. Предполагается, что на контуре питания величина пластового давления постоянна, пласт работает в упругом режиме.

Важное предположение состоит в бесконечной проводимости трещин. Это означает, что на поверхности трещины ГРП давление равно забойному  $p_z$  и избавляет от расчета потока внутри трещины, но в случае трещин с недостаточно большой проводимостью точность расчета снижается. Частным условием служит работа скважины при забойных давлениях выше давления насыщения нефти газом.

Рассмотрим зону дренирования горизонтальной скважины с МГРП. Она представляет собой совокупность прямоугольных секторов и полукружностей на краях, рис.5.



Рис. 5. Геометрия задачи. Расположение зон дренирования трещин на горизонтальной скважине. Образование общей зоны дренирования и ее сегментирование [10]

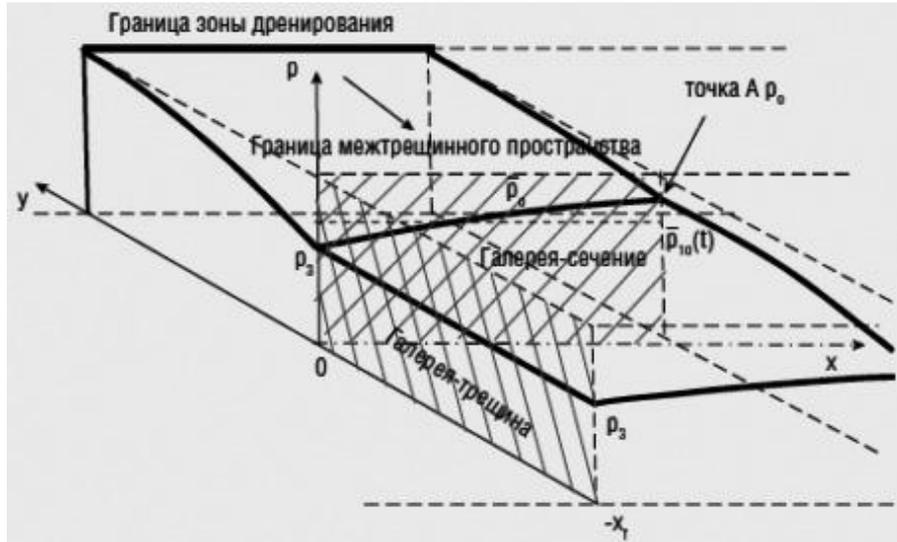


Рис. 6. Геометрия задачи в сегменте II. Стрелкой указан поток к галерее-сечению от границы зоны дренирования [10]

В начальный момент времени давление во всем пространстве вокруг трещины равнозначно пластовому. Затем скважина запускается в работу и давление в трещине резко меняется до забойного. Возмущение начинает распространяться параллельно трещинам до тех пор, пока не достигнет середины между трещинами. Запишем выражение, описывающее динамику притока к нескольким трещинам ГРП горизонтальной скважины для этого периода времени.

Граничные и начальные условия, соответствующие задаче:

- 1)  $p(x,t)=p_n$  при  $t=0$ ;
- 2)  $p(x,t)=p_3$  при  $x=0$ ,  $t>1$ ;
- 3)  $p(x,t)=p_n$  при  $x=x'/2$ ,  $t\geq 1$ ;

где  $x'$  – расстояние между трещинами ГРП, зависящее от числа трещин,  $x' = L/(N-1)$ ;  $L$  – длина скважины между крайними трещинами;  $N$  – число трещин;  $p_3$  – забойное давление,  $p_n$  – пластовое давление.

Выражение для изменения дебита прямоугольного участка дренирования (I) с учетом внешних полукруговых секторов (III) записывается формулой (1):

$$Q_{inD} = \frac{4(N-1)k_x x_f h (p_n - p_3)}{\mu b \sqrt{\pi \kappa t} \cdot \operatorname{erf}\left(\frac{x'}{4 \cdot \sqrt{\kappa t}}\right)} + Q_D, \quad (1)$$

где  $k_x$ ,  $k_y$  – проницаемость пласта по осям  $x$  и  $y$  соответственно,  $x_f$  – полудлина трещины,  $h$  – нефтенасыщенная толщина пласта,  $b$  – объемный коэффициент нефти,  $\kappa$  – пьезопроводность пласта,  $Q_D$  – дебит радиального притока внешних секторов.

Формула (1) справедлива при малых временах и не учитывает взаимное влияние трещин.

Запишем окончательное выражение для полного дебита горизонтальной скважины с МГРП для любого периода времени. Рассмотрим приток к галерее-сечению, на которой давление меняется очень медленно, рис. 6. Галерея – сечение находится на оси  $y$  в начале координат. Пласт ограничен длиной  $l = R - x_f$  ( $R$  – радиус зоны дренирования).

Запишем новые начальные и граничные условия. В данном случае роль галереи играет сечение пласта вдоль границы межтрещинного пространства.

- 1)  $p(y,x,t)=p_n$  при  $t=0$ ;
- 2)  $p(y,x,t) = \bar{p}_0(x,t)$  при  $y=0$ ,  $t>t_0$ ;
- 3)  $\bar{p}_0(x,0) = p_n$  при  $t=0$ ;
- 4)  $p(l,x,t)=p_n$  при  $y=l$  (наличие системы ППД);

где  $\bar{p}_0$  – среднее давление на границе межтрещинного пространства,  $t_0$  – время, в течение которого возмущение достигает середины между трещинами.

Выражение для изменения полного дебита горизонтальной скважины с МГРП для любого периода времени записывается формулой (2):

$$Q_{1D} = \frac{2k_y L h \left( p_n - \frac{p_0}{2} - \frac{p_3}{2} \right) \operatorname{erf}\left(\frac{l}{2 \cdot \sqrt{\kappa t}}\right)}{\mu b \sqrt{\pi \kappa t} \operatorname{erf}\left(\frac{l}{2 \cdot \sqrt{\kappa t}}\right)} + Q_D, \quad (2)$$

$$\text{где } p_0 = \frac{p_n - \left(\frac{1}{2} - A\right) p_3}{1/2 + A},$$

$$A = \frac{(N-1)k_x x_f \operatorname{erf}\left(\frac{l}{2 \cdot \sqrt{\kappa t}}\right)}{k_y L \cdot \operatorname{erf}\left(\frac{l}{4(N-1) \cdot \sqrt{\kappa t}}\right)},$$

$$\operatorname{erf}\left(\frac{l}{2 \cdot \sqrt{\pi k t}}\right) = \frac{2}{\sqrt{\pi}} \int_0^{l/(2 \cdot \sqrt{\pi k t})} e^{-u^2} du,$$

$$Q_D = \frac{2\pi k_y h}{\mu b} \cdot \frac{p_n - p_0}{\ln\left(1 + \frac{\sqrt{\pi k t}}{r_{\text{эф}}}\right)},$$

где  $r_{\text{эф}}$  – эффективный радиус скважины с трещиной ГРП,  $p_0$  – давление, которое возникает на границе межтрещинного пространства.

В формуле для  $Q_D$  не учитывается анизотропия, так как в секторе III пласт изотропный.

## 2. Модель скважины сложной конфигурации

В настоящее время в нефтяной отрасли используются различные типы скважин: вертикальные, вертикальные с трещиной ГРП, наклонно направленные, горизонтальные, горизонтальные с трещинами ГРП, многоствольные горизонтальные, горизонтально-разветвленные, многоярусные и др. Многообразие типов скважин, а также различное вскрытие пласта приводят к появлению большого числа формул для расчета дебита скважин и решения задач неустановившейся фильтрации [11]. Для некоторых сложных случаев аналитические формулы расчета притока отсутствуют.

Для моделирования сложной траектории стволов в горизонтальной плоскости используется набор вертикальных скважин или узлов, достаточно близко расположенных друг к другу, рис. 7. Данный подход позволяет описать течение жидкости к стоку, представляющему собой искривленную вертикальную трещину, проходящую через всю толщину пласта. Дебиты узлов  $Q_i$  в горизонтальной плоскости определяются при решении системы уравнений (3), учитывающей интерференцию узлов.

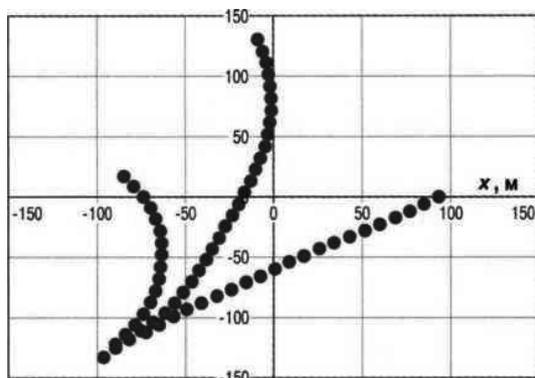


Рис. 7. Моделирование траектории стволов в горизонтальной плоскости при помощи набора узлов [12]

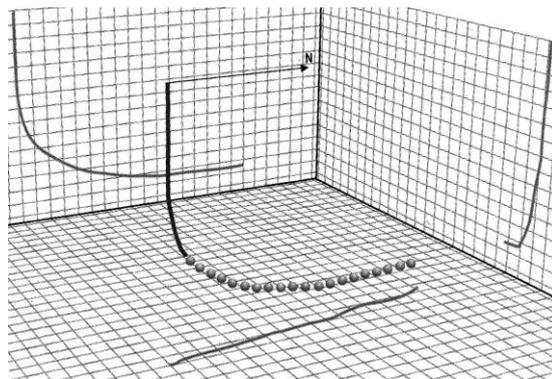


Рис. 8. Представление ствола скважины в пласте набором сфер [12]

$$Q_i = \frac{2\pi\varepsilon}{\ln R_k/r_w} \left( \left[ p_f + \frac{1}{2\pi\varepsilon} \sum_{j=1}^n Q_j \ln \frac{r_{ij}}{R_k} \right] - p_{bi} \right), \quad (3)$$

$$Q_i = \frac{2\pi\varepsilon}{\ln R_k/r_w} \left( \left[ p_f + \frac{1}{2\pi\varepsilon} \sum_{j=1}^n Q_j \ln \frac{r_{ij}}{R_k} \right] - p_{bi} \right),$$

$$Q_n = \frac{2\pi\varepsilon}{\ln R_k/r_w} \left( \left[ p_f + \frac{1}{2\pi\varepsilon} \sum_{j=1}^n Q_j \ln \frac{r_{nj}}{R_k} \right] - p_{bn} \right)$$

где  $Q_i$  – дебит узла,  $\varepsilon$  – гидропроводность пласта,  $r_w$ ,  $R_k$  – радиус соответственно скважины и контура питания;  $p_f$ ,  $p_b$  – соответственно пластовое и забойное давление,  $r_{ij}$  – расстояние между узлами.

Для расчета дебита скважины фильтрационные сопротивления суммируются в вертикальной и горизонтальной плоскостях. Данный подход позволяет учесть любые конфигурации горизонтальных стволов и расположение работающих участков. Задача решена для установившегося и неустановившегося потоков жидкости в пласте. Область применения этого расчетного метода связана только с пластами малой толщины, точнее – с малым отношением толщины пласта к длине горизонтального участка ствола и реализована в ПК Saphir, рис. 9.

Наиболее общее решение получено при использовании трехмерного потока жидкости к скважине [11] путем введения сферического потока к узлам, моделирующим ствол горизонтальной (многоствольной) или любой другой скважины, рис. 8, [12].

В основе предлагаемого метода использовался подход, примененный ранее при решении системы уравнений (3) [13-15], т.е. задача решалась с учетом интерференции отдельных узлов. Только в данном случае узлы представляли собой не вертикальные скважины, как для уравнения (3), а сферы. Для этого выведена система  $n$  уравнений с  $n$  неизвестными дебитами узлов (4-5).

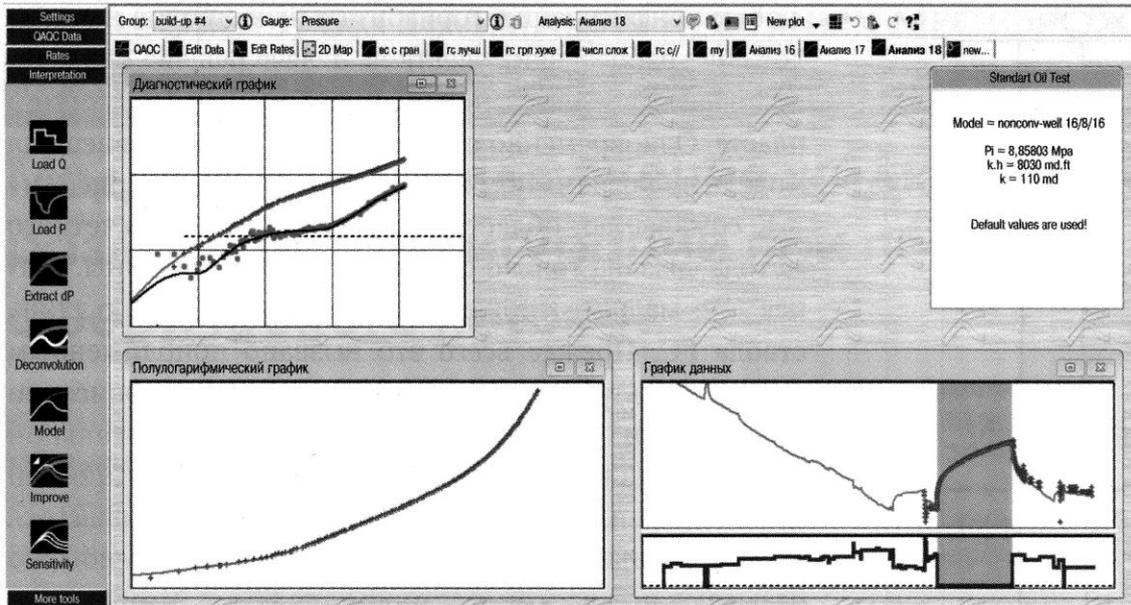


Рис. 9. Пример использования внешней модели для многоствольной скважины (non-conventional well) в ПК Saphir KAPPA Engineering [12]

$$Q_i = - \frac{4\pi\epsilon}{\sum_{m=-x}^{m=x} \alpha_{im}} \left( \left[ p_f - \frac{1}{4\pi\epsilon} \sum_{j=1}^n \left( Q_j \sum_{k=-\infty}^{k=x} \alpha_{1k} \right) \right] - p_{bi} \right); \quad (4)$$

$$Q_i = - \frac{4\pi\epsilon}{\sum_{m=-x}^{m=x} \alpha_{im}} \left( \left[ p_f - \frac{1}{4\pi\epsilon} \sum_{j=1}^n \left( Q_j \sum_{k=-\infty}^{k=x} \alpha_{ik} \right) \right] - p_{bi} \right);$$

$$Q_i = - \frac{4\pi\epsilon}{\sum_{m=-x}^{m=x} \alpha_{im}} \left( \left[ p_f - \frac{1}{4\pi\epsilon} \sum_{j=1}^n \left( Q_j \sum_{k=-\infty}^{k=x} \alpha_{mk} \right) \right] - p_{bi} \right)$$

$$\alpha_{im} = \frac{h}{\sqrt{r_w^2 + (-2hm)^2}} + \frac{h}{\sqrt{r_w^2 + (-2hm - 2Z_{wi})^2}} - \frac{h}{\sqrt{R_k^2 + (-2hm)^2}} - \frac{h}{\sqrt{R_k^2 + (-2hm - 2Z_{wi})^2}}; \quad (5)$$

$$\alpha_{ik} = \frac{h}{\sqrt{R_k^2 + (-2hk)^2}} + \frac{h}{\sqrt{R_k^2 + (-2hk - 2Z_{wi})^2}} - \frac{h}{\sqrt{r_{ij}^2 + (-2hk)^2}} - \frac{h}{\sqrt{r_{ij}^2 + (-2hk - 2Z_{wi})^2}}$$

где  $h$  - толщина пласта;  $Z_w$  - расстояние от подошвы пласта до центра узла.

Как и для предыдущего алгоритма расчета, в данном случае при использовании сферического потока возможно задание различных забойных давлений в узлах, что позволяет учитывать потери на трение и влияние гидрозатворов в скважине с горизонтальным окончанием и, если необходимо, в вертикальных скважинах.

## Заключение

Представлен алгоритм определения количества фактически работающих трещин и параметров пласта в скважинах с МГРП.

Новая модель для анизотропного пласта позволяет количественно прогнозировать дебиты горизонтальных скважин с МГРП в течение работы на упругом режиме.

Представление траектории ствола (стволов) скважины в виде набора узлов и использование метода суперпозиций привели к возможности создания алгоритма расчета установившейся и неустановившейся фильтрации для любых типов скважин с любыми работающими участками.

Это позволяет: а) заменить множество формул, полученных для частных случаев; б) описать приток к скважинам, для которых не существует аналитических формул; в) выбрать наиболее эффективную схему дренажа скважины с учетом особенностей коллектора, рациональной эксплуатации скважины; г) интерпретировать КВД скважин сложной конфигурации; д) снизить трудоемкость расчетов по сравнению с расчетами с использованием конечно-разностного моделирования.

Работа выполнена при поддержке гранта РФФИ 18-07-00503 А.

# New interpretation models overview for well test

S.G. Volpin, A.G. Djachenko, N.P. Efimova, O.V. Lomakina, Ju.B. Chen-len-son

**Abstract:** Article overviews new flow models for well test interpretation, among which multi-fractured horizontal well and complex-architecture well models are of the great interest. Diagnostic plot for multi-fractured horizontal well provided calculation examples and anisotropic reservoir rate calculation formulas. Also provided formulas and calculation examples for complex-architecture well and thin reservoir, steady and transient filtration calculation formulas for any type of wells with different flow intervals configuration.

**Keywords:** well testing, hydraulic fracturing, horizontal well, complex-architecture wells.

## Литература

1. Узе Оливье, Витуря Дидье, Фьярэ Оле. Анализ динамических потоков. Capra Engineering, 2017. 743 с.
2. D.Bourdet et al. 1983. A New Set of Type Curves Simplifies Well Test Analysis. World Oil (May): 95-106.
3. F.M.Giger, L.H.Reiss & A.P.Jourdan. The reservoir engineering aspects of horizontal drilling. In SPE Annual Technical Conference and Exhibition. Texas: Society of Petroleum Engineers, 1984.
4. Fanhui-Zeng, Xiaozhao-Cheng, Jianchun-Guo, Chuan Long & Yubiao-Ke. A New Model to Predict the Unsteady Production of Fractured Horizontal Wells. Sains Malaysiana 2016, 45(10), 1579–1587
5. Z.J.Li, X.L.Pu, G.Wang, Y.Cheng & Y.Su. A novel low-fluorescence anti-sloughing agent for a drilling fluid system and its mechanism analysis. Natural Gas Industry 33: 97-101, 2013.
6. Y.Xiao, T.F.Wang, J.Z.Zhao, Y.Q.Hu & Y.Luo. Computational model of total stress field while multiple fracturing. Oil Drilling & Production Technology 2009(3): 90-93,
7. Yuan, Y.Z., Zhang, L.H., Wang, J. & Pu, Y.W. A binomial deliverability equation for horizontal gas wells in formations with nonlinear seepage flow features. Oil & Gas Geology 2009(1): 122-126.
8. Е.И.Гришина, Н.А.Морозовский, М.И. Кременецкий. Обоснование и прогноз продуктивности горизонтальных скважин на основе комплексных гидродинамических и геофизических исследований // Инженерная практика. – 2016. – №7.
9. Н.А.Морозовский. Методика комплексного геофизического контроля разработки низкопроницаемых коллекторов в условиях искусственной и естественной макротрещиноватости // Диссертация на соискание учёной степени канд. техн. наук. Москва, РГУ нефти и газа имени И. М. Губкина, 2016. — 120 с.
10. С.В.Елкин, А.А.Алероев, Н.А.Веремко, М.В.Чертенков, Р.Ф.Исмагилов. Модель динамики притока жидкости в анизотропном пласте к горизонтальной скважине с МГРП// Инженерная практика. 2017, №12, С. 68-77.
11. E.Ozkan R.Raghavan. New solutions for Well-Test-Analysis Problems: Part 1 - Analytical Considerations 11 SPE Formation Evaluation. - 1991. - V. 6. - № 3. - P. 359-368.
12. В.А. Иктисанов. Моделирование движения жидкости к скважинам различной конфигурации при помощи сферического потока// Нефтяное хозяйство. 2018. №5. С. 52-55.
13. В.А. Иктисанов. Гидродинамические исследования и моделирование многоствольных горизонтальных скважин. - Казань: Плутон, 2007. - 124 с.
14. В.А. Иктисанов. Методы расчета продуктивности и интерпретации КВД для скважин сложной архитектуры // SPE 133477.- 2010.
15. В.А. Иктисанов В.А. Особенности притока к скважинам с горизонтальным окончанием // Нефтяная провинция: сетевое научное издание - 2017. - № 1. - С. 95-126.

# Проблемы моделирования переноса в квазипланарных КНИ КМОП нанотранзисторах

Н.А. Агафонов, Н.В. Масальский, В. В. Осипов, А.В. Родителев

ФГУ ФНЦ НИИСИ РАН, Москва, Россия, E-mail: volkov@niisi.ras.ru

**Аннотация:** Обсуждается возможность эффективного моделирования переноса носителей в квазипланарных КНИ МОП транзисторах. Выполнен анализ разнообразных конструкций транзисторов - от классической планарной схемы до трехмерных мультязатворных структур (двух-, трех- и четырех затворные) с учетом возможной асимметрии затвора и канала транзистора. По результатам исследования сформулированы требования к модели для реализации методики численного расчета переноса носителей в таких устройствах.

**Ключевые слова:** КНИ КМОП транзистор, мультязатворная структура, асимметричный затвор, неравномерно-легированный канал, моделирование

## Введение

Масштабирование размеров транзистора приводит к тому, что его топология постепенно становится трехмерной [1]. Увеличивается напряженность электрического поля и уменьшается количество электронов, участвующих в переносе заряда. Проявляются ранее не учитываемые физические эффекты, которые становятся значимо и трехмерными и технологически зависимыми. Толщина окисла становится сравнимой с толщиной слоя заряда в окисле. С уменьшением толщины окисла становится существенным туннелирование горячих носителей. Это приводит к необходимости использования статистики Ферми-Дирака и уравнений Шредингера и Пуассона [2, 3]. При этом затвор уже нельзя считать плоским и нужно учитывать емкость между боковой поверхностью затвора и истоком/стоком. Топология транзистора начинает сильно влиять на его электрические характеристики, поэтому одна и та же аналитическая модель не может описать его для образцов, выполненных по разным технологиям. Транзисторы, отличающиеся не шириной и длиной канала, а только топологией, могут иметь до 30%-е различие в ВАХ [4-6]. Новые технологии и нетрадиционные конструкции транзисторов требуют разработки новых моделей и новых методик моделирования и способов верификации параметров моделей. В настоящей работе анализируется возможность разработки универсальной

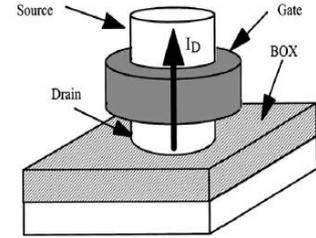
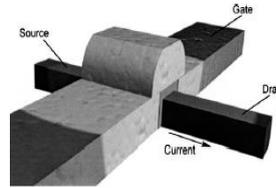
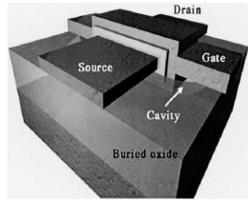
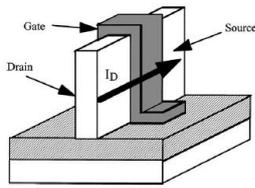
модели квазипланарного транзистора выполненного по технологии «кремний на изоляторе».

Между тем развитие технологии и возрастание сложности объекта моделирования происходят быстрее, чем создаются новые модели для этих технологий. И поэтому с масштабированием размеров транзисторов сложность моделей возрастает быстрее, чем возможности их аналитического описания, а появление новой технологии всегда связано с новыми «неприятностями» в их поведении. Однако, модель не должна досконально следовать за изменениями техпроцесса, поскольку это затрудняет взаимодействие с разработчиками САПР и СБИС [7].

## 1. Конструкции с 3D затвором

Одним из глобальных направлений развития конструкций КНИ МОП транзисторов является переход от классического, планарного, устройства с одним затвором к трехмерным мультязатворным устройствам (двух-, трех- и четырех затворные) [1]. Эти устройства обеспечивают более высокую плотность тока транзистора, чем классический МОП транзистор. Кроме того, они позволяют компенсировать коротко-канальные эффекты (ККЭ): снижение roll-off, DIBL порогового напряжения, крутизны подпороговой характеристики. Новый класс КНИ МОП транзисторов, называемых три плюс затворные устройства несомненно

получит быстрое развитие при решении проблемы расширения технологических



возможностей производства КНИ КМОП транзисторов

Рис. 1. Конструкции 3D КНИ МОП транзисторов [1].

Теоретически, аналитически решив уравнение Пуассона, возможно предсказать толщину кремниевой пленки, которая должна быть в устройствах мультитранзисторного транзистора, чтобы избежать ККЭ или, по крайней мере, чтобы удержать их на требуемом уровне [8]. ККЭ вызваны резким ростом продольного электрического поля у стока со стороны канала транзистора. Таким образом, создаются условия для возникновения демпфирующей области, которая обуславливает усиление механизмов, негативно влияющих на электрические характеристики транзистора [1].

Распределение потенциала в канале КНИ МОП транзистора подчиняется широко известному уравнению Пуассона:

$$\frac{\partial^2 \varphi(x, y, z)}{\partial x^2} + \frac{\partial^2 \varphi(x, y, z)}{\partial y^2} + \frac{\partial^2 \varphi(x, y, z)}{\partial z^2} = - \frac{qN_A(x, y, z)}{\epsilon_s} \quad (1)$$

где  $q$  – заряд электрона,  $\varphi(x, y)$  – потенциал в рабочей области транзистора,  $\epsilon_s$  – диэлектрическая проницаемость рабочей области,  $N_A(x, y, z)$  – концентрация легирования рабочей области.

В общем случае переносом носителей управляют сток и затвор: сток только в латеральном  $x$ -направлении (вдоль канала), затвор – единолично контролирует носителей в  $y$ - и  $z$ -направлениях и конкурирует с изменением электрического поля в  $x$ -направлении из-за напряжения стока. В случае широкого одно- или двух-

затворного транзистора  $\frac{\partial \varphi}{\partial z} = 0$  и (1)

можно записать так:

$$\frac{\partial^2 \varphi(x, y)}{\partial x^2} + \frac{\partial^2 \varphi(x, y)}{\partial y^2} = - \frac{qN_A(x, y)}{\epsilon_s} \quad (2)$$

Для квази двумерного анализа транзистора в рабочей области в  $y$  (вертикальном) направлении распределение потенциала сводится к параболической зависимости [1]. Принятие подобного распределения приводит к 2D распределению потенциала в виде:

$$\varphi(x, y) = \sum_{m=0}^N a_m(x) y^m \quad (3)$$

Рассмотрим квазиклассическую задачу (в приближении зарядового разделения (ПЗР)) нахождения равновесного распределения потенциала в рассматриваемой КНИ структуре [9]. Отметим, что возможность использования приближения ПЗР, означает также и выбор начала отсчета потенциала  $\varphi$ . В общем случае необходимо решать самосогласованную задачу, связанную с нахождением распределения потенциала  $\varphi(x, y)$ , которое будет влиять и на распределение зарядов в рабочей области транзистора. Полагая, что концентрация легирования рабочей области транзистора постоянна по всему ее объему, и пренебрегая влиянием фиксированных окисных зарядов на электростатику канала, распределение потенциала в кремниевой тонкой рабочей области в режиме слабой инверсии может быть записано в виде 2D уравнения Пуассона.

Первоначально рассмотрим случай классического планарного КНИ МОП транзистора. Тогда граничные условия для уравнения Пуассона будут следующие:

$$\varphi(x, 0) = \varphi_f(x) = a_0(x),$$

где  $\varphi_f(x)$  – поверхностный потенциал,

$$\left. \frac{\partial \varphi}{\partial y} \right|_{y=0} = \frac{\varepsilon_{ox}}{\varepsilon_s} \frac{\varphi_f - U_{gs} + U_{FB}}{t_g},$$

где  $U_{gs}$  - напряжение на фронтальном затворе,  $U_{FB}$  - напряжение плоских зон,  $\varepsilon_{ox}$  - диэлектрическая проницаемость подзатворного окисла,  $t_g$  - толщина подзатворного окисла.

Полагая, что толщина погруженного окисла столь большая, что разность потенциалов через любое конечное расстояние в нем незначительна в ортогональном направлении, мы можем

записать  $\left. \frac{\partial \varphi}{\partial y} \right|_{y=0} \cong 0$  в окисле.

Следовательно, обозначая через  $t_s$  толщину рабочей области транзистора, имеем

$$\left. \frac{\partial \varphi}{\partial y} \right|_{y=t_s} = a_1(x) + 2t_s a_2(x) \cong 0,$$

и таким образом  $a_2(x) = \frac{a_1(x)}{2t_s}$ .

Тогда (3) сводится к виду:

$$\begin{aligned} \varphi(x, y) = & \varphi_f(x) + \\ & + \frac{\varepsilon_{ox}}{\varepsilon_s} \frac{\varphi_f - U_{gs} + U_{FB}}{t_g} y - \\ & - \frac{1}{2t_s} \frac{\varphi_f - U_{gs} + U_{FB}}{t_g} y^2 \end{aligned} \quad (4)$$

Подставляя (4) в (3) и полагая  $y=0$ , при котором  $\varphi(x, y) = \varphi_f(x)$ , мы получим уравнение для поверхностного потенциала:

$$\begin{aligned} \frac{\partial^2 \varphi_f(x)}{\partial x^2} - \frac{\varepsilon_{ox}}{\varepsilon_s} \frac{\varphi_f(x) - U_{gs} + U_{FB}}{t_s t_g} = \\ = - \frac{qN_A}{\varepsilon_s} \end{aligned} \quad (5)$$

Вычислив поверхностный потенциал, можно рассчитать и 2D распределение потенциала в канале, используя (4).

Соотношение (5) можно переписать в виде:

$$\frac{\partial^2 \varphi(x)}{\partial x^2} - \frac{\varphi(x)}{l^2} = 0, \quad (6)$$

где

$$\varphi(x) = \varphi_f(x) - U_{gs} + U_{FB} + \frac{qN_A}{l^2},$$

$l$  - характеристическая длина

В результате уравнение Пуассона свелось к обыкновенному дифференциальному уравнению с параметром  $l$ , который определяет распределение потенциала в латеральном направлении. Параметр  $l$  называют «характеристической длиной» транзистора. Он определяет степень влияния ККЭ, свойственного транзисторной структуре, т.е. расстояние проникновения электрического поля из стока в рабочую область. Другими словами уровень контроля, который область стока имеет на область обеднения в канале. Ее значение зависит от толщины подзатворного окисла и толщины кремниевой пленки. Дальнейший анализ и численное моделирование показывают, что эффективная длина затвора МОП транзистора должна быть больше характеристической длины в 5–10 раз, чтобы предотвратить ККЭ и обеспечить приемлемый подпороговый наклон.

В случае двух затворного транзистора отличие в граничных условиях для (2) состоит в следующем [1]:

$$\varphi(x, 0) = \varphi(x, t_s) = \varphi_f(x) = a_0(x).$$

Подставляя эти граничные условия в (3) приходим к уравнению:

$$\begin{aligned} \varphi(x, y) = & \varphi_f(x) + \\ & + \frac{\varepsilon_{ox}}{\varepsilon_s} \frac{\varphi_f - U_{gs} + U_{FB}}{t_g} y - \\ & - \frac{1}{t_s} \frac{\varphi_f - U_{gs} + U_{FB}}{t_g} y^2 \end{aligned} \quad (7)$$

Основное отличие между (7) и (4) то, что в третьем слагаемом исчезла 2. Другими словами, как будто двух затворная структура вдвое тоньше, чем классический транзистор. Характеристическая длина двух затворного транзистора может быть получена тем же путем и ее выражение приведено ниже в таблице. И в случае двух затворной структуры для того, чтобы минимизировать ККЭ, необходимо обеспечить значение параметра  $l$  в том же диапазоне, что и для однозатворного транзистора.

Это понятие можно применить к любым транзисторным конструкциям. Например, для конструкции с полностью

охватывающим затвором и с квадратным поперечным сечением, с учетом того, что

$$\frac{\partial^2 \varphi}{\partial z^2} = \frac{\partial^2 \varphi}{\partial y^2}$$

где располагается максимум электрического поля, тогда уравнение Пуассона имеет вид:

$$\frac{\partial^2 \varphi(x, y)}{\partial x^2} + 2 \frac{\partial^2 \varphi(x, y)}{\partial y^2} = - \frac{qN_A(x, y)}{\epsilon_s} \quad (8)$$

Из приведенного анализа можно сделать следующее обобщение выражений для характеристической длины для разных конструкций транзисторов, а также для разных граничных условий для уравнения Пуассона. В таблице приведены выражения для характеристической длины основных типов конструкций транзисторов обсуждаемых выше.

Таблица 1. Характеристические длины разных конструкций

конструкция	длина	
	Вар 1	Вар 2
один затвор	$l = \sqrt{\frac{\epsilon_s}{\epsilon_{ox}} t_s t_g}$	
двух затворная	$l = \sqrt{\frac{\epsilon_s}{2\epsilon_{ox}} t_s t_g}$	$l = \sqrt{\frac{\epsilon_s}{2\epsilon_{ox}} (1 + \frac{\epsilon_{ox} t_s}{\epsilon_s t_g}) t_s t_g}$
с охватывающим затвором	$l = \sqrt{\frac{\epsilon_s}{4\epsilon_{ox}} t_s t_g}$	$l = \sqrt{\frac{2\epsilon_s t_s^2 (\ln(1 + \frac{2t_g}{t_s}) + \epsilon_{ox})}{16\epsilon_{ox}}}$
двух затворная , асимметричный фронтальный затвор	$l = 1 / (2 \frac{(1 - \frac{C_f}{C_s} + \frac{C_f}{C_b})}{t_s^2 (1 + 2 \frac{C_s}{C_b})})^{1/2}$	

Понятие характеристической длины может использоваться, чтобы оценить максимальную толщину кремниевой пленки (и ширину устройства), которая может использоваться до проявления ККЭ. Рис. 2 показывает максимальную разрешенную толщину канала (рабочей области)  $t_s$ , чтобы избежать ККЭ. Диаграмма получена при помощи соотношений, приведенных в таблице, при толщине подзатворного окисла  $t_{ox}=1.5$  нм. Так, например, для транзистора с длиной затвора 20 нм, толщины кремниевой пленки (канала) для одно затворного полностью обедненного транзистора должна быть в 10 раз меньше чем длина затвора. Для двух затворного устройства толщина рабочей области должна быть уже только в 5 раз меньше длины затвора. Дальнейшее увеличение числа затворов ограничивается структурой полностью охватывающего затвора, где толщина/ширина/диаметр рабочей области может быть равен половине длине затвора. Требования к толщине для трех затворного, Р-затворного и Омега затворного

транзисторов расположены между двух затворным транзистором и транзистором с полностью охватывающим затвором.

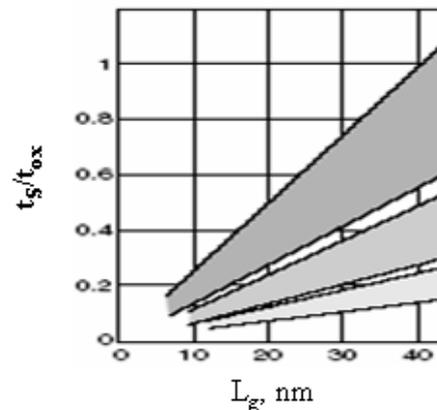


Рис. 2. Диаграмма разрешенных толщин

Использование ультра тонких кремниевых пленок, таким образом, необходимо только для одно затворных устройств. Однако, их применение связано с очень серьезными технологическими проблемами. Поэтому, использование

мульти затворных структур, и особенно с полностью охватывающим затвором существенно снижают технологические требования к рабочей области.

## 2. Конструкция с асимметричным затвором

Повысить эффективность подавления ККЭ призвана транзисторная структура – двух затворный КНИ КМОП транзистор с асимметричным затвором [10].

В данной структуре фронтальный затвор состоит из двух последовательно соединенных материалов М1 и М2 с различными работами выхода.

Такая конфигурация из-за скачка потенциала на границе соприкосновения двух материалов будет обеспечивать одновременное увеличение активной межэлектродной проводимости и подавление ККЭ, по сравнению с двух затворным КНИ КМОП транзистором с затвором из одного материала.

Предположительно, в рассматриваемой структуре пик электрического поля на краю рабочей области у стока будет уменьшаться, что влечет рост усредненного электрического поля под затвором.

В свою очередь это положительным образом скажется на подвижности носителей, обуславливает минимизацию возможности влияния локализованных зарядов на проводимость транзистора, увеличить эффективность управления при помощи затвора общей проводимостью транзистора [11].

Функциональная схема двух затворного КНИ КМОП транзистора с фронтальным затвором, выполненным из двух разных материалов, приведена на Рис. 3.

Фронтальный затвор состоит из двух соединенных материалов М1 (р(+)) поликремния) и М2 (n(+)) поликремния) с длиной  $L_I$  и  $L_{II}$ , соответственно. При этом выполняется условие  $L_I + L_{II} = L_g$ , где

$L_g$  - длина затвора (или длина рабочей области).

Обратный (нижний) затвор выполнен из n(+) поликремния.

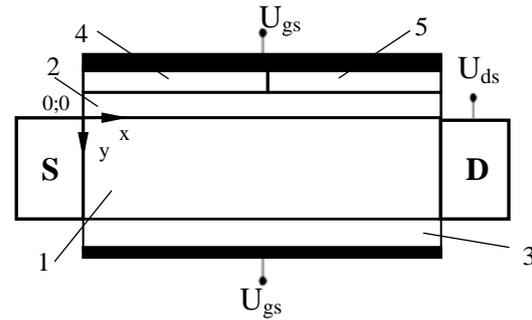


Рис. 3. Схема двух затворного КНИ КМОП транзистора с фронтальным затвором из двух материалов, где 1 – кремниевая рабочая область, 2 – подзатворный окисел фронтального затвора, 3 – подзатворный окисел обратного затвора, 4 – часть фронтального затвора М1, 5 – часть фронтального затвора М2

Так как фронтальный затвор состоит из двух областей, то уравнение Пуассона решается отдельно для каждой. При этом используются следующие граничные условия, с учетом того, что фронтальные напряжения плоских зон для областей I и II (см. Рис. 3) на фронтальном затворе будут различны.

Электрическое поле на границе фронтальный затвор-окисел непрерывно для обеих областей

$$\begin{aligned} \frac{\varepsilon_{ox}}{t_g} (U_{gs} - U_{FB}^{(I)} - \varphi_f^{(I)}(x)) &= \\ = -\varepsilon_s \frac{\partial \varphi^{(I)}(x, y)}{\partial x} \Big|_{y=0} \\ \frac{\varepsilon_{ox}}{t_g} (U_{gs} - U_{FB}^{(II)} - \varphi_f^{(II)}(x)) &= \\ = -\varepsilon_s \frac{\partial \varphi^{(II)}(x, y)}{\partial x} \Big|_{y=0} \end{aligned} \quad (9)$$

где  $\varphi_f^{(I)}(x)$  и  $\varphi_f^{(II)}(x)$  - поверхностный потенциал под областями I и II, соответственно.

Электрическое поле на границе обратный затвор-окисел и обратный канал непрерывно для обоих материалов на фронтальном затворе

$$\begin{aligned} \frac{\varepsilon_{ox}}{t_b} (U_g - U_{FBb} - \varphi(x, t_s)) &= \\ = -\varepsilon_s \frac{\partial \varphi^{(I)}(x, y)}{\partial x} \Big|_{y=t_s} \end{aligned}$$

$$\begin{aligned} \frac{\varepsilon_{ox}}{t_b} (U_g - U_{FBb} - \varphi(x, t_s)) = \\ = -\varepsilon_s \frac{\partial \varphi^{(II)}(x, y)}{\partial x} \Big|_{y=t_s} \end{aligned} \quad (10)$$

где  $t_b$  - толщина подзатворного окисла обратного затвора,  $U_{FBb}$  - напряжение плоских зон на обратном затворе, такое же что и на фронтальном для области II.

Поверхностный потенциал на поверхности двух разнородных материалов затвора на фронтальном затворе непрерывный

$$\varphi^{(I)}(L_1, 0) = \varphi^{(II)}(L_1, 0) \quad (11)$$

Электрическое поле на поверхности двух материалов фронтального затвора непрерывно

$$\frac{\partial \varphi^{(I)}(x, y)}{\partial x} \Big|_{x=L_1} = \frac{\partial \varphi^{(II)}(x, y)}{\partial x} \Big|_{x=L_1} \quad (12)$$

Потенциал на краю рабочей области со стороны истока

$$\varphi^{(I)}(0, y) = U_{bi} \quad (13)$$

Потенциал на краю рабочей области со стороны стока

$$\varphi^{(II)}(L_g, y) = U_{bi} + U_{ds} \quad (14)$$

где  $U_{bi}$  - встроенная разность потенциалов,  $t_s$  - толщина рабочей области транзистора,  $U_{ds}$  - напряжение сток-исток.

При анализе данной структуры тоже используется понятие характеристической длины, ее выражение приведено таблице.

### 3. Неравномерно легированная рабочая область

В данном разделе анализируется вариант асимметричного канала (считая от истока): высоколегированная и низколегированная области. Такая архитектура предназначена для того, чтобы компенсировать влияние таких механизмов: деградация горячих носителей, roll-off порогового напряжения и паразитные биполярные эффекты, сопутствующие для однородно легированных транзисторов [12]. Таким образом, низко легированная область рядом с стоком может обеспечить эффективное

подавление отмеченных выше отрицательных эффектов.

В модельной конфигурации двух затворного симметричного КНИ КМОП нанотранзистора, структура которого схематично представлена на рис. 4, рабочая область состоит из двух частей с разными концентрациями так, что

$$\begin{aligned} L_g &= L_{g_1} + L_{g_2} \\ L_{g_1} &= L_{g_2} \\ N_{A_j} &= \begin{cases} N_{A_1}, 0 \leq y \leq L_{g_1} \\ N_{A_2}, L_{g_2} \leq y \leq L_g \\ N_{A_1} > N_{A_2} \end{cases} \end{aligned} \quad (15)$$

где  $L_g$  - длина рабочей области (длина затвора),  $j = 1$  область с концентрацией легирования  $N_{A1}$  и  $j = 2$  область с концентрацией легирования  $N_{A2}$ .

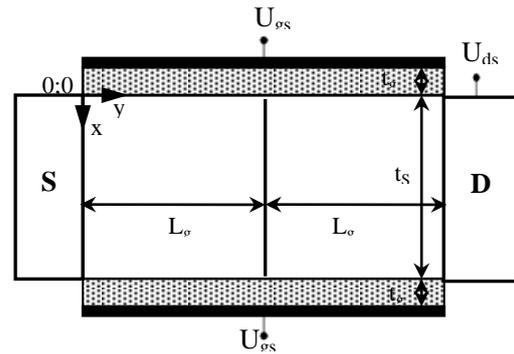


Рис. 4. Структурная схема с неравномерно легированной рабочей областью

Для решения 2D уравнения Пуассона с необходимо дополнить классические граничные условия соотношениями об однородности потенциального и электрического полей на границе концентраций, чтобы учесть свойство неоднородности концентрации легирования рабочей области.

Для решения (2) используются следующие граничные условия в виде:

$$\begin{aligned} \varphi(x, y) \Big|_{x=0} = \varphi_{f_j}(y) \\ \frac{\varepsilon_g}{t_g} (U_{gs} - U_{FB_j} - \varphi_{f_j}(y)) = \\ -\varepsilon_s \frac{\partial \varphi(x, y)}{\partial x} \Big|_{x=0} \end{aligned}$$

$$\begin{aligned} \frac{\varepsilon_g}{t_g} (U_{gs} - U_{FB_j} - \varphi_{b_j}(y)) = \\ - \varepsilon_s \frac{\partial \varphi(x, y)}{\partial x} \Big|_{x=t_s} \\ \varphi_1(x, 0) = U_{bi}^{(S)} \\ \varphi_2(x, L_g) = U_{bi}^{(D)} + U_{DS} \\ \varphi_1(x, L_{g_1}) = \varphi_2(x, L_g - L_{g_1}) = U_{bi}^{(c)} \\ \frac{\partial \varphi_{f_1}(y)}{\partial y} \Big|_{y=L_{g_1}} = \frac{\partial \varphi_{f_2}(y)}{\partial y} \Big|_{y=L_g - L_{g_1}} \end{aligned}$$

где  $\varphi_{f_j}(y)$  - фронтальный поверхностный потенциал  $j$ -той области,  $\varphi_{b_j}(y)$  - поверхностный потенциал на обратной поверхности  $j$ -той области,  $\varepsilon_g$  - диэлектрическая проницаемость подзатворного окисла,  $U_g$  - напряжение на затворах,  $U_{FB_j}$  - напряжение плоских зон,  $U_{bi}^{(S)}$  - встроенная разность потенциалов у истока,  $U_{bi}^{(D)}$  - встроенная разность потенциалов у стока,  $U_{bi}^{(c)}$  - встроенная разность потенциалов на границе разнородно легированных областей,  $U_{DS}$  - напряжение сток-исток.

Распределение потенциала на фронтальной и обратной поверхностях вытекают из решения уравнения Пуассона полученного для параметрического представления потенциала, которое получено в приближении разделения переменных.

В данном случае выражения для потенциалов можно представить в виде:

$$\begin{aligned} \varphi_{f_j}(y) = U_{1j} \exp\left(\frac{y}{l}\right) + U_{2j} \exp\left(-\frac{y}{l}\right) - \\ - A_{ff} l^2 \\ \varphi_{b_j}(y) = U_{3j} \exp\left(\frac{y}{l}\right) + U_{4j} \exp\left(-\frac{y}{l}\right) - \\ - A_{bj} l^2 \end{aligned} \quad (17)$$

$$\text{где } l = t_s \sqrt{\frac{t_g \left(1 + 2 \frac{t_g}{\varepsilon_r t_s}\right)}{2k_s (2t_g + \varepsilon_r t_s)}} \quad \varepsilon_r = \frac{\varepsilon_{ox}}{\varepsilon_s}$$

Коэффициенты  $U_{1j}$ ,  $U_{2j}$ ,  $U_{3j}$  и  $U_{4j}$  определяются из граничных условий.  $k_s$  - подгоночный параметр [12].

Электро-физические характеристики разнообразных транзисторных структур можно анализировать опираясь на решение уравнения Пуассона. Естественно, и это показано выше для каждого конкретного случая необходимы свои граничные условия и свои особенности алгоритма моделирования, что определяется поведением характеристической длины. Это предопределяет возможность разработки универсальной методики моделирования квазипланарных КНИ транзисторных структур с неравномерным легированием. При этом уравнение Пуассона допускает аналитическое решение. Аналитические зависимости обладают неоспоримыми преимуществами перед любым численным решением, а именно – в явной форме выявляют характер зависимости распределения потенциала от существенных физических и топологических факторов. Следует отметить, что аналитическое решение уравнения Пуассона является хорошим приближением и может служить сглаживающей функцией так, где численное решение сталкивается с разного рода «неприятностями».

#### 4. Требования к модели

Физические процессы в КНИ МОП транзисторе описываются следующей фундаментальной системой уравнений [13]:

уравнением Пуассона:  
уравнениями переноса:  
уравнениями непрерывности:

Для строгого расчета переноса носителей через канал транзистора требуется эти уравнения дополнить уравнениями квантовой механики для описания открытых квантовых систем. Предлагается использовать кинетическое уравнение Вигнера, неравновесные функции Грина, квантовое кинетическое уравнение Паули, а также формализм Ландауэра-Бюттикера. Моделирование, основанное на применении уравнения Вигнера, наиболее близко к практическим задачам [14]. Само уравнение Вигнера является полным эквивалентом уравнения Шредингера и записывается для "функции распределения" Вигнера, которая выражается через волновую функцию или соответствующую матрицу плотности. Легко заметить значительное сходство уравнения

Вигнера с кинетическим уравнением Больцмана, точнее, с уравнением Лиувилля, когда интеграл столкновений равен нулю [15].

Полученная система слишком сложна для описания транзисторов. Ее использование для эффективного приборно-технологического моделирования требует некоторых упрощений. Наиболее существенными допущениями являются: деление транзистора на области (исток, сток, область пространственного заряда, область инверсии).

Неизбежное упрощение физических процессов в транзисторе при его моделировании является основной причиной появления дополнительных параметров, а также возможной снижения достоверности результатов. Ниже приведены требования к модели, которые составлены на основании консолидированного мнения большинства экспертов, работающих в этой области:

- масштабируемость
- простота и интуитивная понятность для пользователя
- предсказательная способность
- максимально возможная достоверность;
- физически объяснимое поведение при любых значениях параметров
- широкий диапазон изменения переменных
- вычислительная эффективность (быстрота расчета, сходимость алгоритмов в программах моделирования)
- робастность
- гладкость функций по параметрам
- асимптотическая корректность
- отсутствие корреляции (избыточности) параметров.

«Хорошая» модель должна иметь физически объяснимое поведение при любых сочетаниях физических параметров. При изменении параметров «хорошая» модель должна вести себя точно так, как и объект моделирования. Однако достичь этого очень трудно, поскольку при разработке модели используются упрощающие допущения.

Одним из применений физической модели является предсказание характеристик лабораторного образца транзистора до того, как появится новый и стабильный технологический процесс производства. Поэтому модель должна обладать предсказательной способностью - способностью предсказывать характеристики транзистора при изменении параметров техпроцесса. Эта способность

модели важна для технологов и разработчиков в случаях, когда проектирование техпроцесса настраивается для разработки новых микросхем.

Физичность модели тесно связана с простотой и интуитивной понятностью. Простота и понятность модели для пользователя позволяют избежать большого числа ошибок, которые обычно в изобилии возникают на этапах разработки модели и моделировании микросхем.

Робастность модели — это способность модели правильно отражать поведение реального объекта моделирования. Понятие робастности объединяет понятия вычислительной устойчивости, сходимости вычислительных алгоритмов, способность противостоять ошибкам пользователя, обеспечивать отклик при любых, даже аномальных сочетаниях параметров модели. Для увеличения робастности модель должна иметь гладкие функции и их производные, быть асимптотически корректной и непрерывной по параметрам, в ней должны быть исключены операции деления на нуль, переполнение разрядной сетки, устранены ошибки программирования. Модели, не удовлетворяющим этим условиям, могут дать результаты, не имеющие физического смысла или практически не пригодные.

Работа выполнена при поддержке Программы фундаментальных исследований РАН I.2.П27 НИР № 0065-2018-0113.

## Заключение

Проанализирована возможность эффективного моделирования переноса носителей в квазипланарных КНИ МОП транзисторах.

На основании исследования разнообразных конструкций транзисторов - от классической планарной схемы до трехмерных мультитворных структур (двух-, трех- и четырех затворные) с учетом возможной асимметрии затвора и канала транзистора показана необходимость учета поведения характеристической длины при разработке алгоритма моделирования.

При этом для преодоления «узких» мест следует опираться на аналитическое решение уравнения Пуассона. Сформулированы основные требования к модели для реализации методики численного расчета широкого круга квазипланарных КНИ МОП транзисторов.

# Problems of simulation of transfer in quasi planar SOI CMOS nanotransistors

N.A. Agafonov, N.V. Masalsky, V.V. Osipov, A.V. Roditelev

**Abstract:** The possibility of effective simulation of transfer of carriers in quasiplanar SOI CMOS transistors is discussed. The analysis of various constructions of transistors - from the classical planar circuit to three-dimensional multigate structures (two - three - and four gate) taking into account a possible asymmetry of a lock and the channel of the transistor is made. By results of a research requirements to model for implementation of a technique of numerical calculation of such devices are formulated.

**Keywords:** SOI CMOS transistor, multi gate structure, an asymmetric gate, non-uniform doping channel, simulation

## Литература

1. J.P. Colinge. Multiple-gate SOI MOSFETs. *Solid-State Electronics*. 2004. v. 48. № 3, 897-909.
2. Mamaluy D., Sabathil M., Volg P. Efficient method for calculation of ballistic quantum transport. *Appl. Phys.* 2003. v. 93. 4628-4633.
3. R.Granzner, F.Schwierz, V.Polyakov. An analytical model for the threshold voltage shift caused by two-dimensional quantum confinement in undoped multiple-gate MOSFETs. *IEEE Trans, on Electron Devices*. 2007. v. 54, N 9. 2562-2565.
4. M.Ch, R.Lee, A.Peng, D.Chen, C.Yeh, L.Wu. Miniature RF test structure for on-wafer device testing and in-line process monitoring. *IEEE Trans, on Electron Devices*. 2008. v. 55. 462-465.
5. D.Kim, Y.Jung, M.Park, B.Kim, S.Hong, M.Choi, M.Kang, Y.Yu, D.Whang, S.Hwang. Electrical characteristics of the backgated bottom-up silicon nanowire FETs. *IEEE Trans, on Nanotechnology*. 2008. v. 7, N 6. 683-687.
6. F.Liu, F.He, Y.Fu, J.Ни, W.Bian, Y.Song, X.Zhang, M.Chan. Generic carrier-based core model for undoped four-terminal double-gate MOSFETs valid for symmetric, asymmetric, and independent-gate-operation modes. *IEEE Trans, on Electron Devices*. 2008. v. 55, N 3. 816-826.
7. D.Foty. MOSFET modeling for circuit simulation // *IEEE Circuits and Devices Magazine*. 1998. v. 14, N 4. 26-31.
8. Н.В.Масальский. Характеристики двух затворных КНИ КМОП нанотранзисторов для перспективных технологий с низким уровнем потребляемой мощности. *Микроэлектроника*. 2012. т. 41. № 6, 436-444.
9. Y.Taur, T.Ning. *Fundamentals of modern VLSI devices*. Cambridge University Press. 1998
10. J.Han, C.Kim, Y.Choi. Universal potential model in tied and separated double-gate MOSFETs with consideration of symmetric and separated asymmetric structure. *IEEE Trans. Electron Devices*. 2008. v. 55. № 5. 1472-1480.
11. M.Reyboz, O.Rozeau, T.Poiroux, P.Martin, J.Jomaah. An explicit analytical charge based model of undoped independent Double-Gate MOSFET. *Solid-State Electronics*. 2006. vol. 50. № 8. 1276-1285.
12. Н.В.Масальский. Моделирование распределения потенциала в неоднородно легированной рабочей области двух затворного КНИ КМОП нанотранзистора. *Микроэлектроника*. 2017. т. 46. № 2. 151-160.
13. H.Mattausch, M. Miyake, D.Navarro, N.Sadachika, T.Ezaki, M.Miura-Mattausch, T.Yoshida, S.Hazama. Solving the speed versus accuracy crisis. *IEEE Circuits and Devices Magazine*. 2006. v. 22, N 5. 7-12
14. R.Venugopal, M.Paulsson, S.Goasguen, S.Datta, M.Lunstrom. A simple quantum mechanical treatment of scattering in nanoscale transistor. *J. Appl. Phys.* 2003.v. 93. 5613-5623
15. M.Jilbert, D.Ferry. Resonant behavior and discretedopant effects in narrow ultra short ballistic silicon-on-insulator metal-oxide-semiconductor field effect transistor. *J. Vac Sci. Technol.* 2004. v. 22. 2039-2044.

# Моделирование процессов эволюции популяций конкурирующих автономных агентов

В. Г. Редько

ФГУ ФНЦ НИИСИ РАН, Москва, Россия, E-mail: vgreedko@gmail.com

**Аннотация:** В статье анализируются процессы эволюции в популяциях конкурирующих автономных агентов. Проанализированы три типа моделей. В первой компьютерной модели эволюции рассматривается популяция агентов, которые могут вести борьбу между собой. Причем в процессе мутаций могут исчезать и появляться отдельные сенсоры и эффекторы в нейросетевой системе управления агентов. Показано, что в этой модели «гены агрессивности», ответственные за борьбу между агентами, могут вымирать в процессе эволюционной самоорганизации. Во второй модели рассматривалась эволюция популяций агентов, которые могли как кооперироваться, так и вести борьбу между собой. Проанализированы формы кооперации между агентами в этой модели. В третьей модели рассматривается экономическое сообщество агентов (производителей и инвесторов), которые могут открыто обмениваться информацией между собой. Такой открытый обмен информацией приводит к эффективному сотрудничеству в сообществе агентов.

**Ключевые слова:** автономный агент, нейронная сеть агента, эволюция популяции агентов, устранение конкуренции.

## Введение

В настоящей работе впервые проанализированы особенности процессов эволюции в популяциях конкурирующих автономных агентов (модельных организмов). Рассмотрено три типа моделей эволюции конкурирующих автономных агентов: 1) модель агентов, которые могут вести борьбу между собой, 2) модель агентов, которые могут распознавать себе подобных и создавать коалиции агентов, 3) модель агентов, характеризующая экономические взаимодействия в сообществе инвесторов и производителей.

Проанализированы механизмы устранения конкуренции между агентами.

## Вымирание генов агрессивности в эволюционирующей популяции конкурирующих агентов

В работах [1, 2] исследовалась модель эволюции популяции автономных агентов, характеризующая возникновение естественной разветвленной иерархии целей. В модели рассматривался двумерный

клеточный мир, в клетках могли находиться агенты и их пища (рис. 1).

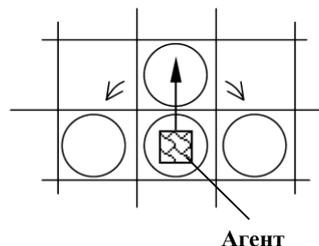


Рис. 1. Агент в двумерной клеточной среде. Агент ориентирован (стрелка показывает направление вперед), кружки – поле зрения агента

Время предполагалось дискретным:  $t = 1, 2, \dots$ . Каждый такт времени агент выполнял одно из следующих действий: 1) питаться, 2) двигаться на одну клетку вперед, 3) повернуть налево, 4) повернуть направо, 5) отдыхать (ничего не делать), 6) делиться (размножиться), 7) ударить агента в клетке впереди (напасть), 8) защищаться.

Каждый агент обладал внутренним энергетическим ресурсом  $R$ , который пополнялся, когда агент выполнял действие «питаться» и расходовался при выполнении других действий. При нападении одного агента на второго (выполнении действия «ударить»)

нападающий агент отнимал ресурс у второго агента, если второй агент не выполнял действие «защищаться»; но если второй агент защищался, то первый агент просто расходовал значительный ресурс.

Система управления агента представляла собой однослойную нейронную сеть, на вход сети подавалась сенсорная информация о ближайшем окружении агента и о внутреннем ресурсе агента, выходы сети определяли действия агента (рис. 2). Входы нейронной сети представляли собой сенсоры (рецепторы) агента, выходы сети – эффекторы агента. Отдельный выход нейронной сети соответствовал одному из возможных действий агента. В данный такт времени агент выполнял то действие, которое соответствовало выходу с максимальным сигналом нейрона. Структура нейронной сети и веса синаптических связей между входом и выходом сети определяли генотип агента. Генотип агента (т.е. его нейронная сеть) в течение жизни агента не менялся. При делении потомок агента наследовал измененную мутациями нейронную сеть агента-родителя. При мутациях модифицировались как веса синаптических связей, так и структура нейронной сети: могли исчезать и вновь появляться отдельные сенсоры или эффекторы. Популяция агентов эволюционировала во времени за счет вариаций структуры и весов синапсов нейронных сетей и отбора тех агентов, которые достаточно быстро набирают ресурс и размножаются.

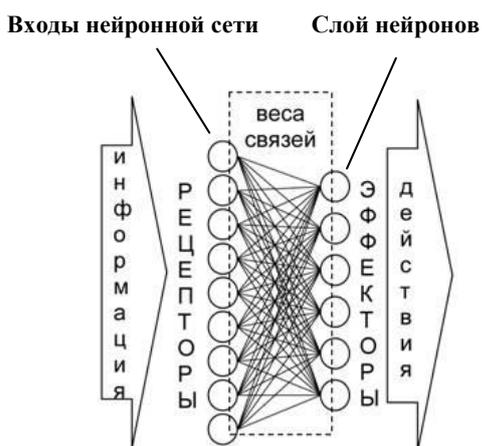


Рис. 2. Структура нейронной сети агента

Модель исследовалась с помощью компьютерных расчетов. В процессе компьютерных экспериментов была показана возможность возникновения иерархии целей

у агентов в процессе эволюции. При этом в некоторых компьютерных экспериментах наблюдались неожиданные пики в зависимостях численности популяции агентов  $N$  от времени  $t$  (рис. 3, нижняя кривая).

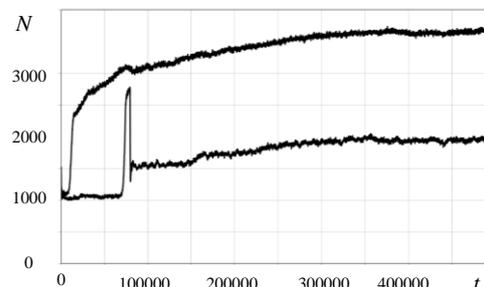


Рис. 3. Зависимость численности популяции  $N$  от времени  $t$  в полной модели (нижняя кривая) и в случае искусственного удаления эффекторов борьбы в системе управления агентов (верхняя кривая)

Анализ системы управления агентов продемонстрировал, что пики соответствовали тем моментам времени, когда у агентов отсутствовали эффекторы борьбы, т.е. в нейронной сети агентов-особей в результате мутаций исчезали эффекторы, ответственные за борьбу между агентами. В эти моменты агенты не тратили свой энергетический ресурс на борьбу с другими агентами, а использовали ресурс для полезных действий. Для подтверждения этого вывода полная модель (с борьбой между агентами) сравнивалась с модифицированной моделью, в которой эффекторы борьбы (т.е. «гены агрессивности») были полностью устранены из системы управления агентов (рис. 3, верхняя кривая). Видно, что в случае искусственного удаления эффекторов борьбы в системе управления агентов установившаяся численность популяции была примерно в 2 раза больше, чем для обычных агентов, ведущих борьбу между собой. Этот эффект связан с тем, что наличие агрессивной конкуренции между агентами приводило к тому, что агенты тратили свой ресурс  $R$  именно на борьбу друг с другом. Этим агентам не хватало ресурса на «созидательную» жизнь, для активного накопления ресурса и расходования его на процессы размножения, приводящие к возрастанию численности популяции. Таким образом, устранение борьбы между агентами, исчезновение «генов агрессивности» приводило к увеличению численности популяции агентов.

Таким образом, гены агрессивности в процессе эволюционной оптимизации систем управления модельных организмов могут вымирать. Хотя не все так просто. Как демонстрирует рис. 3, исчезновение генов агрессивности в полной модели было кратковременным – пики в зависимости численности популяции  $N$  от времени  $t$  были довольно узкие. Кратковременность была связана с тем, что мутационное появление новых агрессивных агентов (с эффекторами борьбы) быстро приводило к тому, что агрессивные агенты вели активную борьбу с «мирными» агентами (не имевшими эффекторов борьбы) и выживали их из популяции. Агрессивная конкуренция приводила к уничтожению мирных агентов. То есть в результате обратных мутаций гены агрессивности восстанавливались, снова начиналась борьба между организмами и численность популяции уменьшалась. Выживали только агенты, имеющие средства для агрессивной конкуренции.

### **Возникновение родственных кланов в эволюционирующей популяции взаимодействующих агентов**

Необходимо отметить, что было очень интересное развитие очерченной модели. В работах [3, 4] в модель были дополнительно введены маркеры (векторы, содержащие несколько компонент), входящие в генотип агентов и характеризующие сходство между агентами. Маркеры не влияли на поведение агента, а служили только как индикаторы сходства между агентами.

Евклидово расстояние между собственным маркером агента и маркерами других агентов подавалось на один из входов нейронной сети агента.

Таким образом, с помощью маркеров агент мог различать сходного с ним агента от существенно отличного от него самого (различать «своего» от «чужого»). При рождении нового агента, участок генотипа, соответствующий маркеру, передавался (с малыми мутациями) от агента-родителя агенту-потомку. В результате потомок был сходен по маркеру с родителем. В итоге в процессе эволюции популяции агентов возникали кланы (виды) сходных между собой агентов. Причем формирование кланов происходило автоматически в процессе эволюционной самоорганизации.

Так как агенты могли бороться между собой, то в модели [3, 4] могли формироваться как кооперация внутри клана агентов-родственников, так и борьба между кланами. При этом автоматически формировалось как разделение на известные типы видов, так и возникали новые типы видов кооперации и конкуренции. Например, формировалось известное разделение на агентов-хищников и агентов-жертв (разделение на «ястребов» и «голубей»). Но, наряду с этим, формировались новые типы конкуренции и кооперации. Например, формировался вид «скворцов», которые коллективно вели борьбу с «ястребами»: «скворцы» жили мирно вблизи друг друга, коллективно довольствуясь сравнительно небольшим питательным ресурсом, собираемым с небольшой территории, но когда к ним приближался хищник, то «скворцы» коллективно атаковали его, защищая свою стаю. Такая стратегия подобна стайной защите от хищника, демонстрируемой многими видами малых стайных птиц.

Отметим, что охарактеризованные сходства взаимодействия между конкурирующими агентами можно использовать и при моделировании социально-экономических систем.

### **Открытый обмен информацией приводит к конструктивной конкуренции между агентами**

Еще одна многоагентная модель исследовалась в работах [5, 6]. Была построена и исследована модель прозрачной экономической системы, состоящей из сообщества агентов-инвесторов и агентов-производителей, которые обмениваются информацией о своих капиталах, намерениях и прибылях, эта информация открыта для всех участников сообщества. Обмен информацией позволяет сформировать децентрализованную систему взаимодействий в рассматриваемом экономическом сообществе. Модель исследована с помощью компьютерного моделирования, результаты которого демонстрируют эффективность предложенной схемы взаимодействий.

В этой модели присутствует конкуренция между отдельными агентами, в результате конкуренции отдельные инвесторы или производители могут вымирать.

Но эта конкуренция достаточно конструктивна, так как имеется открытый обмен информацией между агентами.

Отметим, что эта модель хорошо согласуется с концепцией формирования институтов сотрудничества, которую предлагает известный экономист В.М. Полтерович [7, 8].

## Заключение

В перечисленных моделях намечены подходы к конструктивному взаимодействию внутри популяции агентов, к уменьшению противостояния, к уменьшению конкуренции между агентами.

Можно прямо устранить средства борьбы между агентами (первая модель), возможно

формирование кооперации между отдельными группами агентов (вторая модель), возможен обмен информацией между конкурирующими агентами и формирование конструктивной конкуренции и эффективного сотрудничества в популяции агентов (третья модель).

Отметим, что важность формирования конструктивной конкуренции подчеркивалась в работе [9]. Близкие подходы представлены также в работах [10, 11].

Настоящая работа выполнена в рамках государственного задания по проведению фундаментальных научных исследований (ГП 14) по теме (проекту) «35.16. Моделирование процессов формирования когнитивных способностей автономных агентов». (№ 0065-2018-0002).

# Modeling the evolution of populations of competing autonomous agents

V.G. Red'ko

**Abstract:** The article analyzes the evolutionary processes in populations of competing autonomous agents. Three types of models have been analyzed. The first computer model of evolution considers a population of agents that can fight among themselves. In this model, the individual sensors and effectors could disappear and appear during mutations. It is shown that in this model the “genes of aggressiveness” responsible for the struggle between agents can die out in the process of evolutionary self-organization. In the second model, the evolution of populations of agents that could cooperate and fight with each other was considered. The forms of cooperation between agents have been analyzed in this model. The third model considers the economic community of agents (producers and investors), which can exchange information between themselves. It is shown that open exchange of information between agents can lead to effective cooperation in the community of agents.

**Keywords:** autonomous agent, neural network of the agent, evolution of the population of agents, elimination of competition.

## Литература

1. М.С.Бурцев. Модель эволюционного возникновения целенаправленного адаптивного поведения. 2. Исследование развития иерархии целей. Препринт ИПМ РАН, 2002, № 69.
2. М.С.Бурцев, В.Г.Редько. Влияние агрессии на эволюцию в многоагентной системе. «Проблемы управления безопасностью сложных систем: IX Международная конференция (сборник трудов)». М.: ИПУ, 2002.
3. M.Burtsev, P.Turchin. Evolution of cooperative strategies from first principles. “Nature”, v. 440 (2006), No. 7087, 1041 – 1044.
4. М.С.Бурцев, П.В.Турчин. Эволюция кооперативных стратегий из первых принципов. Приложение С в книге П.В. Турчина «Историческая динамика. На пути к теоретической истории». М.: УРСС, 2007. С. 317 – 328.
5. В.Г.Редько., З.Б.Сохова. Модель взаимодействия инвесторов и производителей в прозрачной экономической системе. «Экономика и математические методы», т. 54 (2018), № 2, 50 – 61.

6. В.Г.Редько., З.Б.Сохова. Исследование процессов самоорганизации в эволюционной модели прозрачной децентрализованной экономики. «Труды НИИСИ РАН», т. 8 (2018), № 2, 103 – 110.
7. В.М.Полтерович. От социального либерализма – к философии сотрудничества. «Общественные науки и современность», (2015), № 4, 41 – 64.
8. В.М.Полтерович. Позитивное сотрудничество: факторы и механизмы эволюции. «Вопросы экономики», (2016), № 11, 5 – 23.
9. В.Г.Редько., М.С.Бурцев, З.Б.Сохова, Г.А.Бесхлебнова. Моделирование конкуренции при эволюции многоагентной системы. Искусственные общества (электронный журнал), т. 2 (2007), № 2, 76 – 89. URL: <http://abm.center/magazine/?ID=44>
10. В.Г.Редько. Будущее России, будущее человечества. «Сложность. Разум. Постнеклассика», 2013, № 3, 55 – 64. URL: <http://cmp.esrae.ru/4-7>
11. В.Г.Редько. Патриотизм или пацифизм. «Тенденции развития науки и образования», (2018), № 38, часть 1, 68 – 71. URL: [http://journal.ru/wp-content/uploads/2016/05/lj31.05.2018\\_p1.pdf](http://journal.ru/wp-content/uploads/2016/05/lj31.05.2018_p1.pdf)

# О нечётких алгоритмах ближайших соседей

А.С. Осипов

ФГУ ФНЦ НИИСИ РАН, Москва, Россия, E-mail: osipa@niisi.ras.ru

**Аннотация:** В статье рассматриваются нечёткие алгоритмы классификации, основанные на правиле «ближайших соседей», в контексте теории алгоритмов распознавания, основанных на вычислении оценок (АВО). Установлено, что нечёткий алгоритм  $K$  ближайших соседей можно отнести к классу АВО. В свою очередь, показано, что стандартные алгоритмы вычисления оценок, зависящие от ряда входных параметров, можно использовать как нечёткие алгоритмы классификации. При этом среди них существуют экстремальные (по входным параметрам) алгоритмы, дающие максимум функционалов качества.

**Ключевые слова:** распознавание образов, теория нечётких множеств, функционалы качества, экстремальные алгоритмы.

## Введение

В последние десятилетия в области распознавания образов и обработки изображений разработано и используется большое число алгоритмов, в той или иной степени использующих элементы теории нечётких множеств [1]. При решении ряда задач компьютерного зрения, в частности, при обработке размытых изображений, использование таких алгоритмов представляется вполне оправданным. Что касается распознавания образов, то, например, классификация объектов, полученных после стадии обработки изображений (в задаче распознавания букв в тексте и т. п.) не всегда является однозначной, и использование нечёткой логики здесь также уместно. Появление «нечетких» алгоритмов вызвало ряд естественных задач, таких как определение областей их применения, поиск оптимальных алгоритмов, предназначенных для решения одной и той же задачи, сравнение этих алгоритмов как между собой, так и с алгоритмами, не использующими нечёткой логики. Эти задачи нашли своё отражение в разрабатываемой в ФГУ ФНЦ НИИСИ РАН методике EDEM сравнительной оценки качества работы алгоритмов обработки и анализа изображений [2-4]. В рамках данной методики алгоритмы тестируются эмпирически на наборе изображений, содержащих различные ситуации, важные для оценки алгоритма. Результаты работы алгоритма сравниваются с заранее известными идеальными решениями (т. н. ground truth образами). Для оценки качества работы алгоритмов на наборе тестовых изображений используются количественные меры, способные, в том числе, оценивать работу некоторых классов нечётких алгоритмов компьютерного зрения (подробнее см. [4]).

Эта методология сравнительного исследования алгоритмов компьютерного зрения находится в соответствии с задачами, сформулированными Ю.М. Журавлевым в 1970-х годах, относительно изучения алгоритмов распознавания образов [5-7]. В частности, как отмечено в [5], «... на первом этапе развития теории и практики распознавания образов возникло большое число различных методов и алгоритмов, применявшихся без какого-либо серьезного обоснования для решения практических задач. Такие методы, как это принято во всех экспериментальных науках, обосновывались непосредственной проверкой - успехом решения реальных задач. Итак, задача состоит в том, чтобы, признав как реальность существование и пользу для практики некорректных процедур решения плохо формализованных задач, изучить с помощью строгих математических методов само множество таких процедур». Можно констатировать тот факт, что к настоящему времени доля эвристических алгоритмов, т. е. алгоритмов, не обоснованных по всем стандартам математической строгости, однако эффективно работающих на некотором количестве эталонных задач лишь возросла. Например, в настоящее время активно применяются различные алгоритмы распознавания образов, использующие нейронные сети. При этом все они «наследуют» открытые вопросы общей теории нейронных сетей: отсутствие строгой теории построения «правильной» нейронной сети, гарантии отсутствия тупиковых ситуаций в обучении сети и т. д. Так что задача развития «строгой теории нестрогих процедур и методов» остаётся актуальной.

В рамках исследования эвристических алгоритмов распознавания образов, в [5-7] была введена в рассмотрение модель алгоритмов вычисления оценок (АВО),

объединяющая ряд известных на тот момент принципов и процедур распознавания. С тех пор изучению АВО было посвящено значительное число работ. Ниже мы остановимся на основных принципах данной модели.

Целью настоящей работы является изучение связи нечетких алгоритмов классификации, где степень принадлежности объекта тому или иному классу обусловлена количеством соседних объектов из обучающей выборки, принадлежащих этому классу, с моделью АВО.

Исследование данной взаимосвязи полезно как для развития нечетких алгоритмов распознавания, так и для развития «нечеткой» компоненты теории эвристических алгоритмов.

Статья организована следующим образом. В разделе 1 преимущественно содержится описание широко используемого нечеткого алгоритма  $K$  ближайших соседей.

Краткая информация, относящаяся к алгоритмам вычисления оценок, содержится в разделе 2.

Следующие разделы посвящены сравнению алгоритмов ближайших соседей с моделью АВО, численным экспериментам и обсуждению результатов.

## 1. Нечеткие алгоритмы ближайших соседей

Прежде всего, напомним несколько основных понятий из теории нечетких множеств.

Именно, пусть  $X$  есть непустое множество (например, множество пикселей изображения). Нечеткое множество  $C$  на  $X$  есть пара  $\langle X, f_C \rangle$ , где  $f_C$  есть отображение  $X$  на  $[0, 1]$ . Значение  $f_C(x)$  для элемента  $x \in X$  называется степенью принадлежности  $x$  множеству  $C$ , а функция  $f_C$  называется функцией принадлежности нечеткого множества. Нечеткое множество называется непустым, если хотя бы для одного элемента  $x \in X$ ,  $f_C(x) > 0$ .

Заметим, что обычные (четкие) подмножества  $M$  из  $X$  включаются в данный подход, если мы будем рассматривать их как стандартные характеристические функции  $1_M : X \rightarrow [0, 1]$ . То есть, например, если у нас имеется пиксель  $x$ , относящийся к некоторому классу (например, сегменту)  $C$ , то в этом случае  $f_C(x) = 1$  и  $f_{C_1}(x) = 0$  для всех классов  $C_1$  отличных от  $C$ .

Будем предполагать, что множество  $X$  конечно.

**Определение 1.** *Нечеткой классификацией  $F$  множества  $X$*

$$F := \langle X, f_{C_1}, \dots, f_{C_l} \rangle, \quad f_{C_m} : X \rightarrow [0, 1], \\ m = 1, \dots, l.$$

*называется совокупность  $l$  нечетких классов, удовлетворяющих условию:*

$$\sum_{m=1}^l f_{C_m}(x) = 1 \quad \forall x \in X. \quad (1.1)$$

Функция  $f_{C_m}$  - степень принадлежности соответствующему классу. Легко видеть, что обычная классификация  $X$  (т. е. разбиение  $X$  на  $l$  непересекающихся подмножеств-классов) является нечеткой классификацией (каждый элемент  $X$  принадлежит ровно одному классу), и в роли функций данной классификации выступают характеристические функции классов.

Обратимся теперь к широко известному нечеткому алгоритму  $K$  ближайших соседей (поиск в google дает около 1900 ссылок на исходную статью [6], содержащую описание алгоритма). Его суть заключается в следующем. Пусть есть множество  $X$ , содержащее объекты  $l$  классов. Каждому объекту из  $X$  сопоставлен вектор его признаков (фиксированной длины). Пусть есть обучающая выборка - конечное множество объектов  $M_R \subset X$ , так что известно, к какому классу относится каждый из элементов  $M_R$ . При этом в данной выборке представлены все  $l$  классов. На первом этапе алгоритма, для каждого элемента  $x \in M_R$  вычисляется степень его принадлежности каждому из  $l$  классов:

$$\mu_j(x) = \begin{cases} 0.51 + 0.49 \left( \frac{n_j}{K_1} \right) & \text{при } i \text{ из класса } j, \\ 0.49 \left( \frac{n_j}{K_1} \right) & \text{в противном случае;} \end{cases} \quad (1.2)$$

(здесь  $i$  - номер элемента в выборке,  $j$  - номер класса,  $K_1$  - некоторое фиксированное число ближайших соседей (близость определяется через расстояние между векторами признаков в некоторой метрике)). Как видно, чем больше у элемента обучающей выборки соседей - одноклассников, тем выше степень его принадлежности своему классу (если соседей  $K_1$ , то степень принадлежности равна 1).

Далее, пусть есть тестовая (или контрольная) выборка  $M_K \subset X$ . Для каждого элемента тестовой выборки  $u$  ищется  $K$  ближайших соседей ( $K$  не обязательно равно  $K_1$ ) и определяются степени его принадлежности каждому классу по формуле:

$$u_j(y) = \frac{\sum_{k=1}^K \mu_{jk}(x_k) \left( \frac{1}{\|y - x_k\|^{\frac{2}{m-1}}} \right)}{\sum_{k=1}^K \left( \frac{1}{\|y - x_k\|^{\frac{2}{m-1}}} \right)}, \quad (1.3)$$

$$j = 1, \dots, l;$$

где  $m$  – фиксированное натуральное число (например, 2),  $C$  – число классов.

В классической версии данного алгоритма [6], элемент  $u$  относят к тому классу, степень принадлежности к которому максимально. Таким образом, данный алгоритм зависит от параметров  $K$ ,  $K_l$  и  $m$ .

Параметр  $m$  отвечает за вклад расстояний от  $u$  до его соседей в вычислении  $u_j(y)$ : чем ближе  $m$  к единице, тем больше вклад ближайших соседей по сравнению с более дальними.

Обозначим данный алгоритм за  $F_K$ . Несмотря на достаточно высокие результаты алгоритма на ряде тестовых задач (ниже мы подробнее остановимся на этом), в [6] отмечались и его недостатки.

Один из основных указанных недостатков заключается в следующем: предположим, элементу  $u$  присвоены степени принадлежности трём классам, равные 0.53, 0.45 и 0.02 соответственно.

Алгоритм  $F_K$  однозначно отнесёт элемент к первому (максимальному) классу. Здравый же смысл подсказывает, что элемент не относится к третьему классу, в то время как его шансы принадлежать к первым двум классам сопоставимы.

В практических приложениях, в частности, в задаче распознавания лиц, обычно используется первый этап данного алгоритма, связанный с вычислением степеней принадлежности для обучающей выборки по формуле (1.2). Они используются для вычисления матрицы разброса данных обучающей выборки (эмпирической ковариационной матрицы). Считается, что тогда матрица будет более «сбалансированной», чем если ее строить без учёта степеней принадлежности. А затем используются вариации метода главных компонент (РСА, подробнее см. [3], [9]).

Помимо нечёткого алгоритма  $K$  ближайших соседей, в [6] была рассмотрена нечёткая версия алгоритма ближайшего прототипа (англ. Fuzzy Nearest Prototype Algorithm). Здесь в тестовой выборке  $M_R$  выбирается  $l$  эталонных прототипов:  $(Z_1, \dots, Z_l)$ , по одному прототипу на класс. Далее, для каждого  $u \in M_K$  определяются степени его принадлежности каждому классу по формуле:

$$u_j(y) = \frac{1}{\|y - Z_k\|^{\frac{2}{m-1}}}, \quad (1.4)$$

$$j = 1, \dots, l;$$

т. е. в данном алгоритме степень принадлежности каждому классу элемента  $u$  зависит от степени его близости к эталонному прототипу. В отличие от нечёткого алгоритма  $K$  ближайших соседей, для эталонных прототипов из  $M_R$  не вычисляются степени принадлежности всем классам, поскольку естественно предполагается, что для эталонного прототипа степень принадлежности своему классу равна 1, и 0 – остальным классам.

Отметим, что оба указанных алгоритма действуют по одной и той же схеме: для каждого объекта по его вектору признаков вычисляются степени принадлежности всем классам. Затем по степеням принадлежности определяется, к какому классу относится объект, т. е. к вектору степеней принадлежности применяется решающее правило.

## 2. Об алгоритмах вычисления оценок

Алгоритмы вычисления оценок, составляющие важную часть теории распознавания образов, основаны на поиске важных частных прецедентов в признаковых описаниях исходных данных. В данных алгоритмах вычисление оценок осуществляется на основе сравнения распознаваемого объекта с эталонными объектами по различным наборам признаков с помощью процедур голосования. При этом, в классической версии модели АВО [5-6], процедура голосования проводится для количественных признаков объектов.

Рассмотрим пример классической версии модели АВО, рассмотренный в [6]. Обозначим алгоритм данной модели буквой  $A$ . Пусть есть множество  $X$ , содержащее объекты из  $l$  классов:  $K_1, \dots, K_l$ . Также, пусть даны конечные множества:  $M_R \subset X$  – решающее множество (обучающая выборка) и  $M_K \subset X$  – контрольное множество (тестовая выборка). Обучающая выборка размера  $m$  содержит  $m_1$  объектов класса  $K_1$ ;  $m_2$  объектов класса  $K_2$ ; ...;  $m_l$  объектов класса  $K_l$ . Решается стандартная задача классификации, т. е. требуется классифицировать объекты  $M_K$ . Каждый

объект  $X$  отождествлён с набором (вектором) его числовых признаков длины  $n$ . Наборы признаков, соответствующие  $M_R$ , располагаются по строкам в виде таблицы: вначале идут объекты  $S_1, \dots, S_{m_1}$ , относящиеся к классу  $K_1$ , затем идут  $S_{m_1+1}, \dots, S_{m_2}$ , относящиеся к классу  $K_2$  и т. д.

Рассмотрим множество  $\{1, 2, \dots, n\}$  и обозначим через  $\{\Omega_k\}$  совокупность всех его подмножеств мощности  $k$ ,  $1 \leq k \leq n$ . Пусть  $\Omega_k \subset \{\Omega_k\}$  и  $\Omega_k = \{r_1, \dots, r_k\}$ . Множества  $\Omega_k$  называют опорными множествами АВО.

Далее, берется произвольный объект  $S \subset M_K$  и его набор признаков  $(\alpha_1, \dots, \alpha_n)$ . Выберем в таблице, задающей множество  $M_R$ , и в строке, задающей  $S$ , координаты с номерами  $r_1, \dots, r_k$  и удалим остальные столбцы в  $M_R$  и координаты в  $S$ . Полученные строки обозначим через  $S(\Omega_k)$   $i=1, \dots, m$  и  $S(\Omega_k)$  соответственно.

Рассмотрим (пороговые) величины  $\varepsilon_1, \dots, \varepsilon_n$ ,  $\varepsilon, \varepsilon_k \geq 0$ ,  $\varepsilon \subset \{0, 1, \dots, k\}$ . Пусть  $S(\Omega_k) = (\alpha_{r_1}, \dots, \alpha_{r_k})$ ,  $S(\Omega_k) = (\beta_{r_1}, \dots, \beta_{r_k})$ , также пусть  $\rho$  - некоторая числовая метрика. Обозначим число невыполненных неравенств  $\rho(\alpha_{r_i}, \beta_{r_i}) \leq \varepsilon_{r_i}, \dots, \rho(\alpha_{r_k}, \beta_{r_k}) \leq \varepsilon_{r_k}$  через  $\sigma$ . Строки  $S$  и  $S_i$  назовем близкими по  $\Omega_k$ , если  $\sigma \leq \varepsilon$ . Пусть  $\{S_i\}^j$  - множество всех строк из таблицы  $M_R$ , близких к  $S$  по множеству  $\Omega_k$ , принадлежащих классу  $K_j$ ,  $j=1, \dots, l$ .

Наконец, введем в рассмотрение величины  $\gamma(S_1), \dots, \gamma(S_m)$ ; характеризующие «важность» элементов обучающей выборки для распознавания. Положим

$$\Gamma_{\Omega_k}^j(S) = \sum_{S_i \in \{S_i\}^j} \gamma(S_i).$$

Далее обозначим

$$\Gamma_j(S) = \sum_{\Omega_k \subset \{\Omega_k\}} \Gamma_{\Omega_k}^j(S). \quad (2.1)$$

Величины  $\Gamma_j(S)$  указывают на близость строки (объекта)  $S$  к элементам класса  $K_j$ , или, в иных терминах,  $\Gamma_j(S)$  - голос в пользу принадлежности  $S$  к классу  $K_j$ .

Пусть  $S_i = (\beta_1, \dots, \beta_n)$  и  $S = (\alpha_1, \dots, \alpha_n)$ . Количество выполненных неравенств из совокупности  $\rho(\alpha_1, \beta_1) \leq \varepsilon_1, \dots, \rho(\alpha_n, \beta_n) \leq \varepsilon_n$  обозначим через  $r(S, S_i)$ . Для вычисления  $\Gamma_j(S)$  в соответствии с (2.1) существует следующая легко выводимая из комбинаторных соображений формула:

$$\Gamma_j(S) = \sum_{i=m_{j-1}+1}^{m_j} \gamma(S_i) \sum_{t=0}^{\varepsilon} C_{r(S, S_i)}^{k-t} C_{n-r(S, S_i)}^t \quad (2.2)$$

Таким образом, в результате описанных выше действий, строка признаков  $(\alpha_1, \dots, \alpha_n)$ , характеризующая объект  $S$ , преобразуется в набор  $(\Gamma_1(S), \dots, \Gamma_l(S))$ . Пусть контрольное множество  $M_K$  состоит из  $q$  объектов:  $S_1, \dots, S_q$ .

Тогда, применительно к  $M_K$ , описанные выше действия можно представить, как преобразование матрицы  $I$  размера  $q \times n$  признаков (описаний) этих объектов

$$I = I(S_1, \dots, S_q) = \begin{pmatrix} \alpha_{11} & \dots & \alpha_{1n} \\ \vdots & \dots & \vdots \\ \alpha_{q1} & \dots & \alpha_{qn} \end{pmatrix},$$

(где  $(\alpha_{p1}, \dots, \alpha_{pk})$  - строка признаков объекта  $S_p$ ,  $p=1, \dots, q$ ) в матрицу  $\Gamma$  размера  $q \times l$  оценок по классам:

$$\Gamma = \Gamma(S_1, \dots, S_q) = \begin{pmatrix} \Gamma_{11} & \dots & \Gamma_{1l} \\ \vdots & \dots & \vdots \\ \Gamma_{q1} & \dots & \Gamma_{qn} \end{pmatrix}, \quad (2.3)$$

где  $\Gamma_{pj} = \Gamma_j(S_p)$ . Оператор  $R$ , переводящий матрицу  $I$  в матрицу  $\Gamma$  называют распознающим оператором. Этот оператор вводится для всех алгоритмов вычисления оценок.

Следующим этапом алгоритма является собственно классификация, осуществляемая при помощи фиксированного решающего правила  $r$ . Приведём его типичный пример (см. [5]-[7]). Пусть  $\Delta_1, \Delta_2$  - заданные положительные величины. Тогда алгоритм заносит объект  $S$  в класс  $j$ , если

- 1)  $\Gamma_j(S) - \Gamma_i(S) \geq \Delta_1, \quad j \neq i;$
- 2)  $\frac{\Gamma_j(S)}{\sum_{i=1}^l \Gamma_i(S)} \geq \Delta_2, \quad j=1..l.$

Если ни для одного номера  $j$  не выполняется одновременно 1) и 2), то алгоритм отказывается от классификации  $S$ . В частности, это происходит, когда все  $\Gamma_j(S)=0$ .

Можно символически записать, что оператор  $r=r_A$  переводит матрицу  $\Gamma$  в информационную матрицу  $Inf_A$  размера  $q \times l$  с элементами из множества  $\{0, 1, Na\}$ , где 1 означает, что элемент выборки отнесён к соответствующему классу, 0 - что не отнесён, символ  $Na$  - что алгоритм отказывается отнести элемент к соответствующему классу (из условия 1) следует, что строки такой матрицы содержат максимум одну единицу). Таким образом, алгоритм  $A$  представляет собой композицию двух операторов: распознающего оператора и решающего правила:

$$A = r_A \circ R_A. \quad (2.4)$$

Также в [6] рассматривалось понятие эффективности алгоритма, выраженное посредством значений функционалов качества. Именно, пусть классы элементов  $M_K$  заранее известны. Положим  $M_K \cap K_j = K_j^*$ ,  $j=1, \dots, l$ . Сопоставим алгоритму  $A$  разбиение  $K_j^*$  на

непересекающиеся подмножества  $K_{jp}$ ,  $p=0, \dots, l$ . В множество  $K_{jp}$  при  $p>0$  отнесем все объекты из  $K_j$ , которые алгоритм  $A$  отнес к классу  $p$ , а в множество  $K_{j0}$  - все элементы  $K_j$ , которые алгоритм отказался классифицировать. Мерой эффективности алгоритма назовем величину

$$\Psi_A^0 = \frac{1}{\mu(M_K)} \sum_{j=1}^l \mu(K_{jj}'), \quad (2.5)$$

где  $\mu(M_K)$  и  $\mu(K_{jj}')$  - меры соответствующих множеств (например, число элементов в них). Это - стандартная статистическая мера оценки качества алгоритма, характеризующая долю правильно классифицированных элементов (подробнее см. [2], [4]).

Если ввести величины  $\gamma_1, \dots, \gamma_l$ ,  $0 \leq \gamma_j \leq 1$ , характеризующие «важность» своих классов, то эффективность можно определить как:

$$\Psi_A^1 = \frac{1}{\mu(M_K)} \sum_{j=1}^l \gamma_j \mu(K_{jj}'). \quad (2.6)$$

Пусть заданы величины  $v_{ij}$ ,  $0 \leq v_{ij} \leq 1$ ;  $i, j=1, \dots, l$ . При  $i$ , отличном от  $j$ , будем называть  $v_{ij}$  штрафом за отнесение элемента из класса  $K_i$  в класс  $K_j$ , а при  $i=j$  назовём  $v_{ii}$  поощрением за правильное распознавание элементов класса  $K_i$ . Пусть также заданы величины  $\varphi_j$ ,  $0 \leq \varphi_j \leq 1$ , которые характеризуют штраф за отказ от распознавания элемента из класса  $K_j$ . В этом случае можно ввести следующую меру эффективности алгоритма [6]:

$$\Psi_A^2 = \max\{\tilde{\Psi}_A^2, 0\}, \quad (2.7)$$

где

$$\tilde{\Psi}_A^2 = \frac{1}{\mu(M_K)} \sum_j v_{jj} \mu(K_{jj}') - \left( \sum_{\substack{i,j \\ i \neq j}} v_{ij} \mu(K_{ij}') + \sum_j \varphi_j \mu(K_{j0}') \right).$$

Пусть  $\{A\}$  есть множество описанных выше алгоритмов распознавания. Алгоритмы этого вида кодируются набором из  $n+m+4$  значений параметров  $k, \varepsilon_1, \dots, \varepsilon_n, \varepsilon, \gamma_1, \dots, \gamma_m, \Delta_1, \Delta_2$  (поскольку полностью ими характеризуется). Следовательно, можно рассматривать меры эффективности (2.5)-(2.7) как функции нескольких переменных, аргументами которой являются эти параметры:

$$\Psi_A^n = \Psi_A^n(k, \varepsilon_1, \dots, \varepsilon_n, \varepsilon, \gamma_1, \dots, \gamma_m, \Delta_1, \Delta_2), \\ n = 0, 1, 2.$$

Для этих параметров можно определить естественные области их изменения. Именно, обозначим через  $T$  множество значений параметров, удовлетворяющих условиям:  $k, \varepsilon$  - целые числа,  $\varepsilon \leq k$ ;  $1 \leq k \leq n$ ;  $0 \leq \varepsilon_p \leq \varepsilon^{\max}$ ,  $p=1, \dots, n$ ;  $0 \leq \gamma_i \leq 1$ ,  $i=1, \dots, m$ ;  $0 < \Delta_1 \leq \Delta_1^{\max}$ ,  $0 < \Delta_2 \leq 1$ .

Для  $\Delta_1^{\max}$  из условия 1) можно вывести следующее значение

$$\Delta_1^{\max} = C_n^k \sum_{i=1}^m \gamma_i.$$

Любым алгоритмом множества  $\{A\}$  каждая строка множества  $M_K$  может быть отнесена к одному из  $l$  классов, либо не классифицирована. Число строк в  $M_K$  конечно. Таким образом, функционалы (2.5)-(2.7), являющиеся функциями на множестве  $T$ , принимают на этом множестве лишь конечное число значений. Тем самым, справедливо следующее утверждение ([6], Теорема 3):

**Утверждение 1.** В точках множества  $T$  достигается  $\sup_T \Psi_A^p$ ,  $p = 0, 1, 2$ .

Алгоритм  $A$ , на котором эта верхняя грань функций эффективности достигается, называется экстремальным. Тем самым, задача построения экстремального алгоритма полностью сводится к отысканию максимумов функций многих переменных. Для отыскания экстремумов могут быть применены методы переборного типа (в случае небольшого числа целочисленных параметров), методы градиентного типа, релаксационные методы [10], метод случайного поиска.

### 3. Нечёткая классификация методом вычисления оценок

Прежде всего, заметим, что меры эффективности (другое их название - функционалы качества [7]) (2.5)-(2.7) принимают значения от 0 до 1. Это даёт возможность определить на  $\{A\}$  нечёткое множество «эффективных алгоритмов»  $Ef_A$ :

$$Ef_A := \langle \{A\}, \Psi_A \rangle,$$

где  $\Psi_A$  - одна из мер (2.5)-(2.7).

Далее, обратимся к матрице (2.3) предположим, что она не содержит нулевых строк. В противном случае, как следует из решающего правила 1)-2), алгоритм  $A$  отказывается отнести объект, соответствующий нулевой строке, к какому-либо классу (чему соответствует строка  $(Na, \dots, Na)$  в матрице  $Inf_A$ ). Введем в рассмотрение матрицу

$$M = M(\Gamma) = (\mu_{pj}(\Gamma))_{p=1, \dots, q}^{j=1, \dots, l}; \\ \mu_{pj}(\Gamma) = \frac{\Gamma_{pj}}{\sum_{k=1}^l \Gamma_{pk}} = \frac{\Gamma_j(S_p)}{\sum_{k=1}^l \Gamma_k(S_p)};$$

т. е. матрица  $M$  получается из матрицы  $\Gamma$  посредством нормирования ее строк. Из Определения 1 можно заключить, что матрица  $M$  представляет собой матрицу нечеткой

классификации выборки  $M_K$ . При этом элемент  $\mu_{pj}(\Gamma)$  есть степень принадлежности объекта  $S_p$  классу  $j$ .

**Определение 2.** Матрица  $\Gamma$  называется матрицей нечеткой классификации оценок.

Что касается решающего правила 1)-2), то оно переписывается в виде

- i.  $\mu_j(S) - \mu_i(S) \geq \Delta'_1, \quad j \neq i;$
- ii.  $\mu_j(S) \geq \Delta'_2, \quad j = 1..l.$

Фиксированные величины  $\Delta'_1$  и  $\Delta'_2$  в этом решающем правиле имеют естественные области изменения:  $0 < \Delta'_1 \leq 1, 0 < \Delta'_2 \leq 1$ . Остальные параметры алгоритма  $A$  изменений не претерпевают.

**Определение 3.** Алгоритм  $F(A)$ , действующий по схеме:

$$F(A): I \xrightarrow{R_{F(A)}} M \xrightarrow{r_{F(A)}} \text{Inf}_{F(A)}$$

называется нечетким представлением алгоритма вычисления оценок  $A$ .

Здесь  $R_{F(A)}$  и  $r_{F(A)}$  есть, соответственно, распознающий оператор и решающее правило алгоритма  $F(A)$ .

Как и алгоритм  $A$ ,  $F(A)$  кодируется набором параметров  $(k, \varepsilon_1, \dots, \varepsilon_n, \varepsilon, \gamma_1, \dots, \gamma_m, \Delta'_1, \Delta'_2)$ . Таким образом, если ввести множество  $F(T)$ , отличающееся от множества  $T$  заменой областей изменения параметров  $\Delta_1$  и  $\Delta_2$  интервалами  $(0,1]$  (т. е. областями изменения  $\Delta'_1$  и  $\Delta'_2$ ), то будет справедливо Утверждение 1 о существовании экстремального (нечеткого) алгоритма на множестве  $F(T)$ .

Таким образом, как нечеткие алгоритмы классификации, семейство  $\{F(A)\}$  обладает следующими достоинствами: вариативностью в построении нечеткой классификации и существованием экстремальных алгоритмов.

Связь алгоритмов вычисления оценок с теорией нечетких множеств отмечалась и в исходных работах по модели АВО. Особенно это касалось стадии распознавания, которой соответствуют операторы  $R_A$ . Так, согласно [11], «Величины  $a_{ij}$  (элементы матрицы оценок (2.3) - А.О.) интерпретируются в терминологии размытых множеств как значение принадлежности  $S$  классу  $K_j$ . Мы будем пользоваться более коротким термином: «оценка  $S$  по  $K_j$ ». Но более подробная информация о такой связи в данных работах отсутствует.

Что касается рассмотренных в разделе 1 нечетких алгоритмов ближайших соседей, то их также можно интерпретировать как алгоритмы модели АВО. Как и алгоритмы

вычисления оценок, они включают в себя две стадии: распознавание и собственно классификация (решающее правило).

Рассмотрим нечеткий алгоритм  $K$  ближайших соседей  $F_K$ , описанный в разделе 1. Пусть  $M_R$  - обучающая выборка объема  $m$ ,  $x_1, \dots, x_m$  - ее объекты, которым сопоставлены векторы (строки) их признаков длины  $n$ ,  $y$  - распознаваемый объект (отождествленный со своим вектором признаков).

Далее, процедура распознавания идет согласно описанной выше процедуре распознавания для алгоритма  $A$  с некоторыми изменениями.

Именно, вместо величин  $\gamma(S_1), \dots, \gamma(S_m)$  определим величины  $\gamma_j(x_i, y)$ ,  $i=1, \dots, m, j=1, \dots, l$ , характеризующие значимость объекта  $x_i$  как элемента класса  $j$  для распознавания объекта  $y$ , по формуле

$$\gamma_j(x_i, y) = \mu_{ij}(x_i) \left( \frac{1}{\|y - x_i\|^{m-1}} \right), \quad (3.1)$$

где  $\mu_{ij}$  вычисляются согласно формуле (1.2), а  $m$  - фиксированное натуральное число (как в формуле (1.3)). Затем таблица  $M_R$  копируется  $l$  раз: вначале идет набор строк  $x_1, \dots, x_m$ , которым соответствуют величины  $\gamma_1(x_i, y)$ , затем такой же набор  $x_1, \dots, x_m$ , которым соответствуют величины  $\gamma_2(x_i, y)$ , и т. д. Далее, пусть расстояние между векторами считается в Евклидовой метрике;  $\delta = \sqrt{\delta_1^2 + \dots + \delta_n^2}$  - радиус окрестности  $y$ , содержащей  $K$  ближайших соседей из выборки  $M_R$ . Тогда, если в процедуре распознавания для алгоритма  $A$  положить:  $k=n, \varepsilon=0$  (учитывается близость по всем признакам),  $\varepsilon_1 = \delta_1, \dots, \varepsilon_n = \delta_n$ , приходим к следующему аналогу формул (2.1)-(2.2):

$$\Gamma_j(y) = \sum_{k=1}^K \gamma_j(x_k, y) \equiv \sum_{k=1}^K \gamma_{jk}; \quad (3.2)$$

здесь  $x_k$  - ближайшие соседи  $y$ .

Правая часть этой формулы есть в точности числитель в правой части формулы (1.3).

Решающее правило для данного алгоритма отличается от приведенного выше решающего правила 1)-2), однако оно является одним из стандартных решающих правил модели АВО [5] (более простым, чем 1)-2). Именно, алгоритм заносит объект  $y$  в класс  $j$  если

$$\Gamma_j(y) - \Gamma_i(y) > 0, \quad j \neq i; \quad i = 1, \dots, l;$$

В случае невыполнения данного условия для всех  $j$ , алгоритм отказывается от классификации  $y$ .

Алгоритм, описанный по приведённой процедуре, обозначим через  $A_K$ . Таким

образом, если в алгоритме  $F_K$  считать  $m$  и  $K_j$  фиксированными параметрами, то алгоритм  $A_K$  можно закодировать следующим набором параметров:  $(\varepsilon_1, \dots, \varepsilon_n, \gamma_{11}, \dots, \gamma_{1K}, \dots, \gamma_{11}, \dots, \gamma_{1K})$ , всего их  $n+IK$ .

Первая группа параметров характеризует зависимость алгоритма от  $K$ . Можно предположить, что заданы интервалы изменения всех этих параметров, тогда можно ввести множество  $T_K$ , представляющее собой прямое произведение этих интервалов.

Повторяя рассуждения доказательства Утверждения 1, о существовании экстремальных алгоритмов, приходим к следующему результату.

**Утверждение 2.** *Существует алгоритм  $A_K$  модели АВО, идентичный нечеткому алгоритму  $K$  ближайших соседей. В точках множества  $T_K$  достигается  $\sup_{T_K} \Psi_A^p$ ,  $p = 0, 1, 2$ .*

Для знаменателя правой части формулы (1.3) имеем, пользуясь (1.1) и (3.1):

$$\begin{aligned} \sum_{k=1}^K \frac{1}{\|y - x_k\|^{m-1}} &= \sum_{k=1}^K \sum_{j=1}^l \mu_{jk} \frac{1}{\|y - x_k\|^{m-1}} = \\ &= \sum_{j=1}^l \sum_{k=1}^K \mu_{jk} \frac{1}{\|y - x_k\|^{m-1}} = \sum_{j=1}^l \Gamma_j(y). \end{aligned}$$

Сопоставляя полученное соотношение с (3.2), приходим к следующему заключению.

**Утверждение 3.** *Алгоритм  $F_K$  есть нечеткое представление алгоритма вычисления оценок  $A_K$ :  $F(A_K) = F_K$ .*

Построения, подобные изложенным выше, можно провести и для нечеткого алгоритма ближайшего прототипа.

## 2. Численные эксперименты

В данном разделе мы эмпирически сравним производительность алгоритмов семейства  $\{F(A)\}$  и нечетких алгоритмов из раздела 1 ( $K$  ближайших соседей и ближайшего прототипа). В качестве тестовой задачи берется классическая задача распознавания ирисов из базы Iris Фишера [12]. Данная база многократно применялась в различных исследованиях. В частности, она использовалась в [8] при тестировании упомянутых выше нечетких алгоритмов. Эта база содержит данные о 150 экземплярах ириса, по 50 экземпляров из трёх видов — Iris setosa, Iris virginica и Iris versicolor. Для каждого экземпляра измерялись четыре характеристики (в сантиметрах): 1. Длина чашелистика, 2. Ширина чашелистика, 3. Длина лепестка, 4. Ширина лепестка. Один из

классов (Iris setosa) линейно отделим двух остальных.

В статье [8] распознавание проводилось по правилу «откладывания одного» (leave one out).

Согласно этому правилу, из базы брался один объект для распознавания в качестве тестового, а остальные объекты (в данном случае 149) составляли обучающую выборку. Такая процедура распознавания применялась к каждому элементу базы.

Таким образом, в данных экспериментах менялась обучающая выборка.

Помимо базы Iris, в [8] также рассматривалась база Iris23, состоящая из объектов второго и третьего класса, всего 100 объектов (эти объекты линейно неразделимы).

В нечетком алгоритме  $K$  ближайших соседей варьировались параметры  $K$  и  $K_j$  (из формулы (1.2)), параметр  $m$  полагался равным 2. Таблица 1 содержит результаты данного распознавания: количества неправильно классифицированных объектов базы Iris в зависимости от значений этих параметров.

Таблица 1.

Количества неверно распознанных объектов базы Iris в зависимости от значений  $K$  и  $K_j$  нечеткого алгоритма  $K$  ближайших соседей.

$K_j \backslash K$	1	3	5	7	9
1	6	6	6	6	6
2	6	6	6	6	6
3	5	5	5	5	6
4	5	5	5	5	5
5	4	4	5	5	5
6	4	4	4	4	4
Среднее	5	5	5.2	5.2	5.3

Что касается нечеткого алгоритма ближайшего прототипа, в [8] для тестируемого объекта в качестве ближайшего прототипа каждого из трёх классов использовался ближайший к нему объект соответствующего класса.

Этот алгоритм требует существенно меньше действий по сравнению с алгоритмом  $K$  ближайших соседей. Использовались вычисления как с исходными векторами объектов, содержащими все 4 признака, так и с векторами, содержащими лишь 2 признака (признаки 3. и 4.). Результаты классификации объектов баз Iris с исходными и двумерными векторами признаков представлены в таблицах 2 и 3 в виде матриц неточностей (англ. confusion matrix). Правильно классифицированные объекты относятся к элементам матрицы, находящимся на главной диагонали, неправильно классифицированные — ко всем остальным.

Таблица 2.  
Результаты распознавания объектов базы Iris нечетким алгоритмом ближайшего прототипа по четырем признакам объектов.

Классы	1	2	3
1	50	0	0
2	0	45	5
3	0	7	43

Таблица 3.  
Результаты распознавания объектов базы Iris нечетким алгоритмом ближайшего прототипа по третьему и четвертому признакам.

Классы	1	2	3
1	50	0	0
2	0	48	2
3	0	4	46

Как видно из этих таблиц, в случае распознавания лишь по двум последним признакам результаты классификации лучше, чем по полному набору, что свидетельствует в пользу использования классификации по различным поднаборам признаков, реализованной в модели АВО.

Результаты распознавания объектов базы Iris23 были хуже, чем в случае исходной базы Iris, что объяснимо, учитывая неразделимость между объектами двух классов в Iris23.

Для алгоритмов семейства  $\{F(A)\}$  параметры, определяющие множество  $F(T)$ , выбирались следующим образом:  $\varepsilon=0$ ;  $k \subset \{2,3,4\}$ ;  $\gamma_1 = \dots = \gamma_m = 1$  (все объекты обучающей выборки одинаково важны для распознавания),  $\Delta_1=0.1$ ,  $\Delta_2=0.6$ .

Средние расстояния между признаками в базе Iris следующие:  $\varepsilon_{1m}=1.8$ ,  $\varepsilon_{2m}=1.2$ ,  $\varepsilon_{3m}=2.95$ ,  $\varepsilon_{4m}=1.2$ .

Соответственно, выбиралось по 3  $\varepsilon_i$ , равномерно распределенные на интервалах  $0 \leq \varepsilon_i \leq \varepsilon_{im}$ ,  $i=1, \dots, 4$ .

Итого получилось  $36=3*12$  алгоритмов. Они тестировались на базах Iris и Iris23.

Приведем результаты распознавания для алгоритмов  $F1$  и  $F2$ , закодированных наборами параметров (4, 0.6, 0.4, 1, 0.4, 0, 1, 1, 1, 1, 0.1, 0.6) и, соответственно, (2, 1.2, 0.8, 2, 0.8, 0, 1, 1, 1, 1, 0.1, 0.6).

Для базы Iris результаты содержатся в Таблицах 4-5.

Таблица 4.  
Результаты распознавания объектов базы Iris алгоритмом  $F1$

Классы	1	2	3
1	49	0	1
2	0	46	4
3	0	1	49

Таблица 5.  
Результаты распознавания объектов базы Iris алгоритмом  $F2$ .

Классы	1	2	3
1	50	0	0
2	0	48	2
3	0	2	48

Как видно, производительность алгоритмов  $F1$  и  $F2$  на базе Iris сопоставима с производительностью нечеткого алгоритма  $K$  ближайших соседей и выше, чем у алгоритма ближайшего прототипа.

Что касается базы Iris23, результаты распознавания ее объектов и для алгоритмов семейства  $\{F(A)\}$  оказались ниже, чем их результаты для базы Iris. Однако они сопоставимы с результатами распознавания для нечетких алгоритмов ближайших соседей.

Описанные выше тесты проходили при меняющейся обучающей выборке. Также проводились эксперименты по распознаванию при постоянной обучающей выборке. В качестве последней, в базе Iris были случайно выбраны по 20 экземпляров каждого класса. Остальные 90 экземпляров составляли тестовую выборку. В этих экспериментах выбиралось по 10 параметров  $\varepsilon_i$ , равномерно распределенные на интервалах  $0 \leq \varepsilon_i \leq \varepsilon_{im}$ ,  $i=1, \dots, 4$ . Остальные параметры оставались такими же, как и в предыдущих тестах. В итоге тестировалось  $3*40=120$  алгоритмов. Наилучшие результаты показали алгоритмы, закодированные наборами параметров (4, 0.9, 0.24, 0.6, 0.12, 0, 1, 1, 1, 1, 0.1, 0.6) и (2, 0.36, 0.24, 0.3, 0.12, 0, 1, 1, 1, 1, 0.1, 0.6)

Непосредственная проверка показала, что в первом случае один объект класса 2 был ошибочно отнесен к классу 3 и наоборот, один объект третьего класса оказался отнесен ко второму. Во втором же случае, два объекта второго класса были ошибочно отнесены к третьему. Здесь для ранжирования этих алгоритмов можно использовать меру (2.6), учитывающую априори заданную «важность» соответствующих классов.

## 4. Заключение

Полученные результаты показали возможность практического использования нечеткого представления алгоритмов модели АВО в качестве алгоритмов классификации.

Эти алгоритмы не требуют вычисления степеней принадлежности классам для элементов обучающей выборки, вариативны в построении нечёткой классификации распознаваемых объектов и допускают существование локально экстремальных алгоритмов.

К их недостатку можно отнести сложность в определении границ изменения расстояний между признаками, однако естественно предположить, что распознаваемый объект находится близко от обучающей выборки.

В таком случае, для определения этих границ можно использовать элементы обучающей выборки (как это делалось в изложенных выше примерах).

Что касается определения набора параметров, характеризующих оптимальный алгоритм, даже в случае небольшого количества параметров аналитический их поиск - сложная задача [10]. Методы случайного поиска (особенно, если алгоритм оценивается с помощью нескольких функционалов качества) представляются более перспективными.

Как уже отмечалось в разделе 1, на практике в нечетких алгоритмах классификации обычно используется лишь стадия распознавания, т. е. вычисление степеней принадлежности классам для распознаваемых объектов. В качестве решающего правила используются достаточно сложные алгоритмы. В этой связи приобретает важность исследование качества выполнения стадии распознавания нечеткими алгоритмами, что в частности отмечалось в [4]. В этой связи представляется оправданным использование нечетких мер близости, предложенных в данной работе.

Работа выполнена в рамках НИР № 0065-2018-0004 («Развитие методов мат. моделирования распределенных систем и соответствующих методов вычисления»).

## On the fuzzy nearest neighbor algorithms

A.S. Osipov

**Abstract:** In the paper some fuzzy classification algorithms based upon a nearest neighbor decision rule are considered in terms of the theory of pattern recognition algorithms based on computation of estimates (AEC). It is shown that the fuzzy  $K$  nearest neighbor algorithm can be assigned to the AEC class. In turn, it is found that some standard algorithms of AEC class, depending on a number of input parameters, can be used as fuzzy classification algorithms. Yet among them there exist extremal algorithms which provide maximum values of their performance measures.

**Keywords:** performance evaluation, pattern recognition, fuzzy set theory, performance measures.

### Литература

1. J.C. Bezdek, J. Keller, R. Krisnapuram, N. Pal. Fuzzy models and algorithms for pattern recognition and image processing. The handbooks of fuzzy set series. vol. 4, 2006, Springer Science & Business Media.
2. П.П. Кольцов, А.С. Осипов, А.С. Куцаев, А.А. Кравченко, Н.В. Котович, А.В. Захаров. О количественной оценке эффективности алгоритмов анализа изображений. Компьютерная Оптика, Т. 39, No 4, 2015, стр. 542-556..
3. А.С. Осипов. Об использовании элементов нечеткой логики в оценке алгоритмов идентификации лиц. Труды НИИСИ РАН, 2016, Т. 6, №2, стр. 62-69.
4. А.С. Осипов. Нечёткие меры и их использование в оценке алгоритмов компьютерного зрения. Труды НИИСИ РАН, Т. 7, №1, 2017, стр. 46-57.
5. Ю.И. Журавлев, В.В. Никифоров. Алгоритмы распознавания, основанные на вычислении оценок. Кибернетика, №3, 1971, стр. 1-11.
6. Ю.И. Журавлев. Экстремальные задачи, возникающие при обосновании эвристических процедур. Проблемы прикладной математики и механики. 1971, стр. 67-75.
7. Ю.И. Журавлев. Об алгебраическом подходе к решению задач распознавания и классификации. Проблемы кибернетики, выпуск 33, М.: Наука, 1978, стр. 5-68.
8. J.M. Keller, M.R. Gray, J.A. Givens. A fuzzy  $K$  – nearest neighbor algorithm. IEEE transactions on systems, man, and cybernetics, 1985, vol. SMC-15, №4, pp. 580-585.
9. K.C. Kwak, W. Pedry. Face recognition using a fuzzy fisher classifier, Pattern Recognition vol. 38, №10, 2005, pp. 1717–1732.
10. В.В. Рязанов. Оптимизация алгоритмов вычисления оценок по параметрам, характеризующим представительность эталонных строк. Журнал Вычислительной Математики и Математической Физики, Т. 16, №6, 1976, стр. 1559-1570.
11. Ю.И. Журавлев. Избранные научные труды. М.: Магистр, 1998, 420с.
12. База ирисов Фишера. Доступна по адресу: <http://archive.ics.uci.edu/ml/datasets/Iris>

# Оценка погрешности приближенного решения с применением ассоциированных полиномов

Д. В. Гулуа<sup>1</sup>, Е.К. Гребенникова<sup>2</sup>

<sup>1</sup>Грузинский технический университет, Тбилиси, Грузия, <sup>2</sup>ФГУ ФНЦ НИИСИ РАН, Москва, Россия,

E-mail's: <sup>1</sup> [mssf4477@ya.ru](mailto:mssf4477@ya.ru), <sup>2</sup> [ekaterina.grebennikova@inbox.ru](mailto:ekaterina.grebennikova@inbox.ru)

**Аннотация:** Неявная четырёхслойная полудискретная схема для абстрактного эволюционного уравнения сведена к двухслойным схемам с применением алгоритма возмущений. На основе этих схем строится приближённое решение исходной задачи. В гильбертовом пространстве получена оценка для погрешности приближённого решения с помощью ассоциированных полиномов.

**Ключевые слова:** полудискретная схема, алгоритм возмущений, ассоциированные полиномы.

Известно, что большой интерес представляют вопросы, связанные с построением и исследованием алгоритмов приближённого решения эволюционных задач, в том числе реализация и исследование многослойных схем для решения этих задач. Основная трудность, которая возникает при реализации многослойных схем (особенно для многомерных задач), состоит в необходимости использовании большой оперативной памяти, которая возрастает пропорционально росту числа слоев. На наш взгляд одной из наиболее эффективных возможностей преодоления этой проблемы является использование алгоритма возмущений для расщепления многослойных схем. Алгоритм возмущений, применительно к разностным схемам для дифференциальных уравнений, был рассмотрен в работе [1]. Отметим, что предложенный алгоритм близок к методам, рассмотренным в работах Г.И.Марчука и В.В.Шайдунова [2] и В.Перейра [3, 4].

В работах [5], [6] реализация чисто неявной трёхслойной схемы для эволюционного уравнения с помощью алгоритма возмущений, сводится к реализации двухслойных схем. В указанных работах в банаховом пространстве с помощью полугрупп доказывается явная оценка для погрешности приближённого решения исходной задачи при достаточно общих предположениях относительно данных задачи.

В представленной работе, в гильбертовом пространстве, для приближённого решения задачи Коши для эволюционного уравнения с самосопряжённым положительно определённым оператором, рассматривается чисто неявная четырёхслойная полудискретная схема, которая сводится к двухслойным схемам. С помощью решений этих схем, строится приближённое решение исходной задачи. Следует отметить, что первая двухслойная схема даёт при-

ближённое решение с точностью первого порядка, а решение каждой последующей схемы уточняет предыдущее решение на порядок. Для оценки погрешности приближённого решения, применяется подход, предложенный в [7], где исследование устойчивости линейных многослойных методов основано на свойствах класса полиномов от многих переменных (т.н. ассоциированные полиномы). Эти полиномы являются естественным обобщением классических полиномов Чебышева второго рода.

## Сведение неявной четырёхслойной схемы к двухслойным схемам.

В Гильбертовом пространстве  $H$  рассмотрим следующую эволюционную задачу:

$$\frac{du(t)}{dt} + Au(t) = f(t), \quad t \in [0, T] \quad (1)$$

$$u(0) = u_0, \quad (2)$$

где  $A$  – самосопряжённый положительно определённый оператор в  $H$  с областью определения  $D(A)$  всюду плотной в  $H$ ;  $f(t)$  – непрерывно дифференцируемая функция со значениями из  $H$   $u_0$  – заданный вектор из  $H$ ;  $u(t)$  – искомая функция.

Введём на  $[0, T]$  сетку  $t_k = k\tau$ ,  $k = 0, 1, \dots, n$ , с шагом  $\tau = T/n$ . Для аппроксимаций первой производной воспользуемся чисто неявной четырёхслойной схемой. В результате, уравнение (1) в точке  $t = t_k$  можно представить в следующем виде:

$$\frac{u(t_k) - u(t_{k-1})}{\tau} + Au(t_k) + \frac{\tau \Delta^2 u(t_{k-2})}{2\tau^2} + \frac{\tau^2 \Delta^3 u(t_{k-3})}{3\tau^3} = f(t_k) - \tau^3 R_k(\tau, u), \quad (3)$$

где  $k = 3, \dots, n$ ;  $\Delta u(t_{k-1}) = u(t_k) - u(t_{k-1})$ ,  $R_k(\tau, u) \in H$ .

Из (3) с помощью алгоритма возмущений [1], получим следующую систему уравнений:

$$\frac{u_k^{(0)} - u_{k-1}^{(0)}}{\tau} + Au_k^{(0)} = f_k, \quad f_k = f(t_k), \quad u_0^{(0)} = u_0,$$

$$k = 1, \dots, n \quad (4)$$

$$\frac{u_k^{(1)} - u_{k-1}^{(1)}}{\tau} + Au_k^{(1)} = -\frac{1}{2} \frac{\Delta^2 u_{k-2}^{(0)}}{\tau^2}, \quad k = 2, \dots, n \quad (5)$$

$$\frac{u_k^{(2)} - u_{k-1}^{(2)}}{\tau} + Au_k^{(2)} = -\frac{1}{2} \frac{\Delta^2 u_{k-2}^{(1)}}{\tau^2} - \frac{1}{2} \frac{\Delta^3 u_{k-3}^{(0)}}{\tau^3}, \quad k = 3, \dots, n \quad (6)$$

Введем следующее обозначение:

$$u_k = u_k^{(0)} + \tau u_k^{(1)} + \tau^2 u_k^{(2)}, \quad k = 3, \dots, n \quad (7)$$

Пусть вектор  $v_k$  является приближённым значением точного решения задачи (1), (2) при  $t = t_k$ ,  $u(t_k) \approx u_k$ . В схеме (5) стартовый вектор  $u_1^{(1)}$  определяем из равенства  $v_2 = u_2^{(0)} + \tau u_2^{(1)} + \tau^2 u_2^{(2)}$ , где  $u_2^{(0)}$  и  $u_2^{(1)}$  находим по схеме (4) и (5) соответственно, а  $v_2$  – приближенное значение  $u(t_2)$  с точностью  $O(\tau^3)$ .

Можно показать, что построенное решение  $v_k$  удовлетворяет следующему уравнению:

$$\frac{\frac{11}{6}v_k - 3v_{k-1} + \frac{3}{2}v_{k-2} - \frac{1}{3}v_{k-3}}{\tau} + Av_k = f_k + \tilde{R}_k(\tau), \quad k = 5, \dots, n, \quad (8)$$

где для невязки  $R_k(\tau)$ , для достаточно гладких начальных данных, справедлива оценка  $\|\tilde{R}_k(\tau)\| \leq c\tau^3$ ,  $c = \text{const} > 0$ ,  $k = 5, \dots, n$  (9)

Исследование схемы (4)–(7) опирается на ряд фактов о полиномах, ассоциированных с разностным уравнением высокого порядка.

### Полиномы ассоциированные с разностным уравнением высокого порядка.

Результаты, приведенные в этом пункте, представляют, на наш взгляд, независимый интерес и нашли применение в исследовании некоторых многослойных схем (см. [7]).

Рассмотрим разностное уравнение порядка  $q \geq 1$  следующего вида

$$z_k - x_1 z_{k-1} - \dots - x_q z_{k-q} = 0, \quad k \geq q, \quad (10)$$

где  $x_i$  ( $i = 1, \dots, q$ ) – действительные числа.

**Лемма 1.** Для решения уравнения (10) справедлива формула

$$z_{k+q-1} = U_k z_{q-1} + (x_2 U_{k-1} + \dots + x_q U_{k-q+1}) z_{q-2} + (x_3 U_{k-1} + \dots + x_q U_{k-q+2}) z_{q-3} + \dots + x_q U_{k-1} z_0, \quad k = 1, 2, \dots,$$

где полиномы  $U_k(x_1, \dots, x_q)$  удовлетворяют рекуррентному соотношению:

$$U_k(x_1, \dots, x_q) = x_1 U_{k-1}(x_1, \dots, x_q) + x_2 U_{k-2}(x_1, \dots, x_q) + \dots + x_q U_{k-q}(x_1, \dots, x_q), \quad k = 1, 2, \dots, \quad (11)$$

$$U_0 = 1, U_{-1} = \dots = U_{1-q} = 0.$$

Лемма доказывается методом математической индукции.

Полиномы  $U_k(x_1, \dots, x_q)$  мы называем полиномами, ассоциированными с разностным уравнением (10). Заметим, что  $U_k(2x, -1)$  ( $q = 2$ )-полиномы Чебышева второго рода. Поэтому естественно назвать полиномы, определяемые рекуррентным соотно-

шением (11), полиномами Чебышева от  $q$  переменных.

**Лемма 2.** Справедлива формула

$$\lambda_i^{n+q-1} = \lambda_i^{q-1} U_n + \lambda_i^{q-2} (x_2 U_{n-1} + \dots + x_q U_{n-q+1}) + \dots + \lambda_i (x_{q-1} U_{n-1} + x_q U_{n-2}) + x_q U_{n-1}, \quad n = 1, 2, \dots, \quad (12)$$

где  $\lambda_i$  ( $i = 1, \dots, q$ ) – корни характеристического уравнения

$$\lambda^q - x_1 \lambda^{q-1} - \dots - x_{q-1} \lambda - x_q = 0. \quad (13)$$

Доказательство опирается на тот факт, что разностное уравнение (для простоты рассмотрим случай  $q = 3$ )  $z_{k+2} - x_1 z_{k+1} - x_2 z_k = 0$ , эквивалентно системе уравнений

$$\begin{aligned} z_k - \lambda_1 z_{k-1} &= z_k^1, \\ z_{k+1}^1 - \lambda_2 z_k^1 &= z_{k+1}^2, \\ z_{k+2}^2 - \lambda_3 z_{k+1}^2 &= 0, \end{aligned} \quad (14)$$

где  $\lambda_i$  ( $i = 1, 2, 3$ ) – корни уравнения (13).

Решение системы (14) при  $z_1 = \lambda_1 z_0$ ,  $z_2 = \lambda_1^2 z_0$  задается формулой

$$z_k = \lambda_1^k z_0 \quad (k = 1, 2, \dots) \quad (15)$$

С другой стороны, в силу леммы 1  $z_{k+2} = U_k z_2 + (x_2 U_{k-1} + x_3 U_{k-2}) z_1 + x_3 U_{k-1} z_0$ . Отсюда при  $z_1 = \lambda_1 z_0$ ,  $z_2 = \lambda_1^2 z_0$  и из (15) следует (12) для  $q = 3$ .

Очевидно, что за  $\lambda_1$  мы можем взять любой корень уравнения (13). Следовательно, формула (12) верна для любого  $i$  ( $i = 1, 2, 3$ ).

Справедлива следующая теорема.

**Теорема 3.** Пусть  $M$  – некоторое множество точек  $E_q$  ( $P = (x_1, \dots, x_q)$ ,  $E_q$  – евклидово пространство размерности  $q$ ). Предположим, что выполняется следующее условие: для любого  $P \in M$  все корни уравнения (13), за исключением, быть может, одного, принадлежат одному и тому же кругу, лежащему внутри единичного круга, а исключительный корень принадлежит единичному кругу. Тогда справедливо соотношение  $\sup_n \{ \sup_{P \in M} |U_n(P)| \} < +\infty$ .

Доказательство теоремы опирается на расщепление уравнения в виде (14) и лемму 1.

### Априорная оценка погрешности приближённого решения.

Для погрешности  $z_k = u(t_k)$  имеем:

$$\frac{\frac{11}{6}z_k - 3z_{k-1} + \frac{3}{2}z_{k-2} - \frac{1}{3}z_{k-3}}{\tau} + Az_k = f_k + r_k(\tau), \quad k = 5, \dots, n, \quad (16)$$

где  $r_k(\tau) = -(\tau^3 R_k(\tau, u) + \tilde{R}_k(\tau))$ .

Учитывая (9) заключаем, что если решение задачи (1), (2) достаточно гладкая функция, то  $\|r_k(\tau)\| = O(\tau^3)$ .

Справедлива следующая теорема.

**Теорема 4.** Пусть  $A$  самосопряжённый положительно определённый оператор в  $H$ .

Тогда справедлива оценка

$$\|z_{k+2}\| \leq c (\|z_2\| + \|z_3\| + \|z_4\| + \tau \sum_{i=3}^k \|z_{k+2}(\tau)\|), \quad (17)$$

где  $c = \text{const} > 0$ ,  $k = 3, \dots, n-2$ .

Выделим основные этапы доказательства теоремы. Из (16), имеем

$$z_{k+1} = \frac{18}{11}Lz_k - \frac{9}{11}Lz_{k-1} + \frac{2}{11}Lz_{k-2} - \frac{6}{11}\tau Lr_{k+1}(\tau), \quad (18)$$

где  $L = (I + \frac{6}{11}\tau A)^{-1}$ .

Введя обозначения  $L_1 = \frac{18}{11}L$ ,  $-L_2 = -\frac{9}{11}L$ ,  $L_3 = \frac{2}{11}L$ ,  $g_{k+1} = \frac{6}{11}\tau Lr_{k+1}(\tau)$ , равенство (18) примет вид:

$$z_{k+1} = L_1 z_k + L_2 z_{k-1} + L_3 z_{k-2} + g_{k+1}.$$

Поэтому по индукции получаем [7]:

$$z_{k+2} = U_{k-2} z_4 + (L_2 U_{k-3} + L_3 U_{k-4}) z_3 + L_3 U_{k-3} z_2 + \sum_{i=3}^k U_{k-i} q_{i+2}, \quad (19)$$

где  $U_k(L_1, L_2, L_3)$  операторные полиномы, которые определяются рекуррентным соотношением:  $U_k(L_1, L_2, L_3) = L_1 U_{k-1} + L_2 U_{k-2} + L_3 U_{k-3}$ ,  $U_0 = I$ ,  $U_{-1} = U_{-2} = 0$ .

Рассмотрим характеристическое уравнение, ассоциированное с разностным уравнением (18) ( $x \in S_p(L) \subset [0, 1]$ )

$$\lambda^3 - \frac{18}{11}x\lambda^2 + \frac{9}{11}x\lambda - \frac{2}{11}x = 0. \quad (20)$$

Доказывается, что для любого  $x \in [0, 1]$  действительный корень уравнения (20) находится в единичном круге, а остальные два корня комплексно-сопряжённые и принадлежат одному и тому же кругу, лежащему внутри единичного круга. Тогда, в силу теоремы 3, операторные полиномы  $U_k(L_1, L_2, L_3)$  ограничены в совокупности. При доказательстве указанного результата использован известный факт, что норма операторного полинома, когда аргумент – самосопряжённый ограниченный оператор, равен  $C$  – норме соответствующего скалярного полинома на спектре этого оператора [8].

Заметим, что для достаточно гладких начальных данных, легко доказываются оценки:

$$\|u(t_k) - v_k\| = O(\tau^3), k = 3, 4. \quad (21)$$

Окончательно, из (17) и (21) следует следующая теорема.

**Теорема 5.** Пусть  $A$  самосопряжённый положительно определённый оператор в  $H$  и решение задачи (1), (2) достаточно гладкая функция. Тогда, если  $\|u(t_k) - v_k\| = O(\tau^3)$ ,  $k = 1, 2$ , то справедлива оценка

$$\|u(t_k) - v_k\| = O(\tau^3), k = 3, \dots, n.$$

## Estimates for the approximate solution error using the associated polynomials

D.V. Gulua, E.K. Grebennikova

**Abstract:** Using the perturbation algorithm, purely implicit four-layer semi-discrete scheme for an abstract evolutionary equation is reduced to two-layer schemes. The approximate solution of the initial problem is constructed under these schemes. Using the associated polynomials, in Hilbert space the estimate for the approximate solution is obtained.

**Keywords:** semi-discrete scheme, perturbation algorithm, associated polynomials.

### Литература

1. В.И. Агошков, Д.В. Гулуа. Алгоритм возмущений для реализации конечномерных аппроксимаций задач. М.: ОВМ АН СССР, 1990. 35 с.
2. G.I. Marchuk, V.V. Shaidurov. Difference Methods and Their Extrapolations. N.Y.; B.; Heidelberg; Tokyo: Springer, 1983. 334 p.
3. V. Pereyra. On improving an approximate solution of a functional equation by deferred corrections // Numer. Math. 8(1966). № 3. P. 376–391.
4. V. Pereyra Accelerating the convergence of discretization algorithms // SIAM J. Numer. Anal. 1967. V. 4. № 4. P. 508–533.
5. Дж. Л. Рогава, Д.В. Гулуа. Алгоритм возмущений для реализации конечно-разностной аппроксимации абстрактной эволюционной задачи и явная оценка погрешности приближенного решения // Доклады Академии Наук, 2014, т. 456, № 4, с. 405–407.
6. Д.В. Гулуа, Дж. Л. Рогава. Об алгоритме возмущений полудискретной схемы для эволюционного уравнения и оценке погрешности приближенного решения с помощью полугрупп // Ж. вычисл. матем. и матем. физ., 2016, т. 56, № 7, с. 1299–1322.
7. Дж. Л. Рогава Полудискретные схемы для операторных дифференциальных уравнений. Тбилиси: Техн. Ун-т, 1995. 288 с.
8. Л.В. Канторович, Г.П. Акилов. Функциональный анализ. М.: Наука, 1977. 741 с.

# Аффинные преобразования и координатизация (аксиоматика) для модулей над общими кольцами

Мзия Цхомелидзе

Грузинский технический университет, Тбилиси, Грузия, E-mail: mziacxomelidze@mail.ru

**Аннотация:** Статья рассматривает вопрос об адаптации алгоритмов распознавания на основе аффинных преобразований и координатизации для модулей над общими кольцами. Адаптация алгоритмов распознавания к аффинным графическим изображениям дает возможность решить важную задачу автоматизации производства модулей общих кругов, а именно гиперплоидных передач, и компенсации влияния указанных искажений за счет использования современных компьютерных технологий.

**Ключевые слова:** аффинное преобразование, распознавание, графическое изображение, модули, концентрические кольца

Аффинными преобразованиями графических изображений называют такие преобразования, при которых точки переходят в точки, прямые в прямые, параллельные прямые в параллельные прямые и сохраняется инцидентность точек и прямых. Аффинным преобразованиям относятся поворот, смещение и масштабирование графических изображений [1 – 4].

Аффинные преобразования и соответствующие искажения графических изображений почтовых индексов составляют главное препятствие в построении модулей общих кругов.

Если при создании модели, которая имеет значительную массу и нестандартные размеры, достичь такого фиксированного положения не удастся, а, следовательно, возникает проблема обеспечения инвариантности распознавания относительно сдвигов, поворотов и масштабирования в плоскости, перпендикулярной к оптической оси считывателя.

Для компенсации аффинных искажений обычно определяют их параметры и выполняют обратные аффинные преобразования, в результате которых графические изображения набирают свой первоначальный вид.

Перспективной является замена обратных аффинных преобразований адаптацией алгоритмов распознавания к аффинным преобразованиям моделей общих кругов.

Основные трудности такой адаптации связаны с отличием представления прямой линии на непрерывной и дискретной плоскостях (рис. 1).

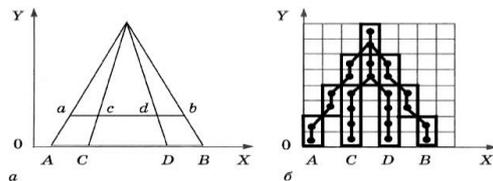


Рис. 1. Падение прямых линий на непрерывных (а) и дискретных (б) площадях

Как следует из рис. 1, при подаче на непрерывной плоскости количество точек, образующих отрезки  $ab$  и  $AB$  (континуум), одинакова, тогда как на дискретной плоскости эта закономерность нарушается. Это означает, что любым точкам  $C$  и  $D$  отрезка  $AB$  на непрерывной плоскости взаимно однозначно соответствуют точки  $c$  и  $d$  отрезка  $ab$ , а на дискретной плоскости такого взаимно однозначного соответствия нет.

Примеры изображения двух наклонных линий I и II на непрерывной и дискретной плоскостях приведены на рис. 2, где  $\varphi_1$  и  $\varphi_2$  - угол наклона соответственно линии.

Как видим наклонная линия на дискретной плоскости имеет вид ступенчатой линии, в которой длины ступеней могут отличаться на единицу.

Изображения линии на дискретной плоскости в дальнейшем будем называть ее трассой, а количество дискретных элементов, из которых эта трасса состоит – ее дискретной длиной.

Трассу отрезка прямой линии, наклоненной под углом  $\varphi < 45^\circ$ , дискретная длина которой определяется проекцией этого отрезка на ось абсцисс, будем называть трассой типа X, а трассу отрезка прямой линии, наклоненной под углом  $\varphi < 45^\circ$ , дискретная длина которой определяется

проекцией этого отрезка на ось ординат, по трассе типа В.

Трасса, которая на рис. 2 соответствует углу наклона  $\varphi_1 = \arctg 8/13 < 45^\circ$  является трассой типа X, а та, что соответствует углу наклона  $\varphi_2 = \arctg 13/8 > 45^\circ$  - по трассе типа В. Заметим, что поскольку  $\varphi_2 = 90^\circ - \varphi_1$ , дискретные длины обеих трасс совпадают и равны 13.

Заметим, что дискретная длина линии при ее повороте от 0 до  $45^\circ$  пропорциональна косинусу угла поворота, а при повороте от  $45^\circ$  до  $90^\circ$  - синусу этого угла (рис. 3). При этом считают, что максимальные значения достигаются при углах поворота 0 и  $90^\circ$ , а минимальное - при угле поворота  $45^\circ$ .

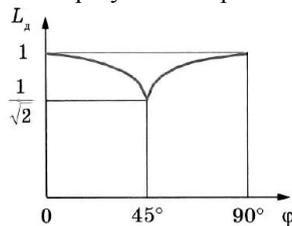


Рис. 2. График изменений дискретной длины линии в зависимости от угла поворота

В случае изображения на дискретной плоскости любой линии существует только восемь направлений переходов между двумя ее соседними элементами (рис. 3).

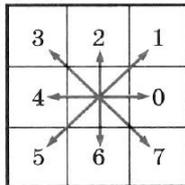


Рис. 3. Возможные переходы между соседними элементами линии

Основные задачи адаптации алгоритмов распознавания к аффинным преобразованиям графических изображений заключаются в определении:

1. Определение параметров аффинных искажений.
2. Определение трасс переходов между элементами графических изображений на дискретной плоскости.

#### Определение параметров аффинных искажений графических изображений

Параметры аффинных искажений можно определить, проанализировав взаимное расположение и расстояния между специальными метками (реперами), напечатанными на почтовых отправлениях или на адресных ярлыках прямоугольной формы, на них наклеиваются.

Задача выделения меток возникает также при поиске лицевой стороны почтовых отправлений. Если при отсутствии аффинных

искажений (письменная корреспонденция) меток удобно использовать прямоугольные стартовые и центровочной метки, то в случае их наличия (посылки) как метки целесообразно использовать геометрические фигуры, инвариантные к поворотам графических изображений.

Таковыми фигурами являются круги или концентрические кольца. Остановимся на последних, поскольку именно метки в виде концентрических колец (в зависимости от их ширины и взаимного расстояния) достаточно просто можно идентифицировать.

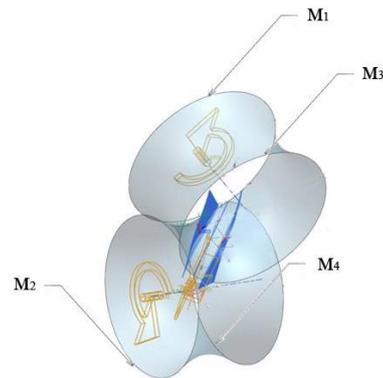


Рис. 4. Общий вид модуля общих кругов для модуля гиперплоидного колеса

Принимая во внимание необходимость проверки попадания адресного ярлыка в зону считывания индекса, целесообразно применить четыре метки, центры которых расположены в четырех точках модуля колесной пары зубчатой передачи (рис. 4).

В представленном на рисунке 4 фрагменте имитационного кинематического формообразования боковых поверхностей зубьев гиперплоидного зубчатого колеса воспроизводилось за координированное вращение гиперблоида по геометрическим характеристикам в представленном случае соответствующее единичному передаточному отношению.

Каждая метка содержит два концентрических кольца и зазор между ними, которые при пересечении линией диаметра образуют последовательность семи черных Ч и белых Б сегментов Ч1 Б2 Ч3 Б4 Ч5 Б6 Ч7 (рис. 5).



Рис. 5. Общий вид метки

Чтобы можно было использовать общий алгоритм выделения меток, у них должны быть одинаковые диаметры, а это возможно при условии равенства суммарного количества элементов.

В зависимости от количества элементов, образующих указанные сегменты, они могут быть широкими Ш, С средними или узкими В, причем количество элементов в этих сегментах удовлетворяет пропорцию 3:2:1.

В связи с этим за условные длины сегментов возьмем Ш = 3, С = 2, В = 1. Тогда для выполнения равенства суммарного количества элементов в рассматриваемых сегментах необходимо, чтобы суммы их условных длин равнялись шести.

Последовательность длин сегментов Ч1 Б2 Ч3 образует код диаметра метки. Всего таких кодов существует семь:

- код 1 - ВСШ - 123,
- код 2 - ВШС - 132,
- код3-СВМ-213,
- код 4 -ССС -222,
- код 5-ПЕНИЕ-231,
- код 6 - ШВС - 312,
- код7-ШСВ-321.

Идентифицируются метки следующим образом: метке М1 присвоен код диаметра 1, метке М2 код диаметра 2, метке М3 - коды диаметров 3, 4, 5, метке М4 -коды диаметров 6, 7. Благодаря этому, кроме идентификации меток, за комбинацией кодов диаметров меток М3 и М4 можно идентифицировать метки.

Чтобы убедиться в правильности расположения меток и отсутствия так называемых проективных искажений графического изображения (обусловленных неперпендикулярностью оптической оси считывателя к плоскости графического изображения), при наличии которых параллельные прямые воспринимаются как сходящиеся или расходящиеся, выполняют проверку длин проекций сторон адресного ярлыка на оси координат (рис. 6):

$$\begin{cases} |x_1 - x_2| = |x_3 - x_4|, \\ |y_1 - y_2| = |y_3 - y_4|, \\ |x_1 - x_3| = |x_2 - x_4|, \\ |y_1 - y_3| = |y_2 - y_4|, \end{cases}$$

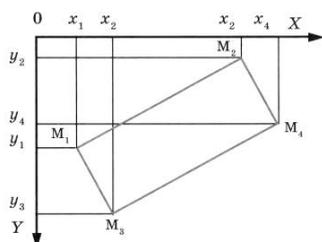


Рис. 6. Определение длин проекций сторон адресного ярлыка на оси координат

С помощью специального алгоритма определяют параметры аффинных искажений графического изображения:

1. Выделение меток.
2. Выделение кодов меток диаметров.
3. Идентификация меток.
4. Определение угла поворота изображения.
5. Возведение произвольного угла поворота изображения к повороту.
6. Определения смещения изображения.
7. Определение масштаба изображения.

Фиксируя последовательности сегментов, которые отвечают требованиям относительно диаметров меток, вычисляем длины L1, L2, L3, L4, L5, L7, семи текущих сегментов каждой строки графического изображения, начиная с черного. Если в строке менее чем семь сегментов, то в нем поиск меток не выполняется.

При наличии в строке семи и более сегментов проверяют возможность предоставления диаметров меток L1=L7, L2 = L6, L3 = L5). Когда указанные условия выполняются, вычисляют суммарную длину D всех сегментов, которая должна удовлетворять неравенству:

$$D_{\min} \leq D \leq D_{\max}$$

Найденную таким образом последовательность сегментов принимают за диаметр метки и вычисляют координаты ее центрального элемента.

Определение кодов диаметров меток сводится к анализу длин сегментов Ч1 Б2 Ч3, что образуют коды диаметров меток.

Проверка правильности расположения меток сводится, как уже отмечалось, к проверке равенств соответствующих проекций сторон адресного ярлыка на оси координат.

Параметры аффинных искажений графического изображения определяются по координатам центральных элементов и диаметрами меток. При этом считается, что поворот и сдвиг графического изображения вследствие его аффинных искажений осуществляется относительно центрального элемента метки М1 (рис. 7).

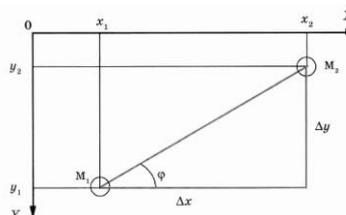


Рис. 7. Определение угла поворота изображения за взаимным расположением центральных элементов меток

Из рис. 10 имеем:

$$\Phi = \text{arctg} (\Delta y / \Delta x)$$

На практике угол поворота удобнее всего определять непосредственно по приращениям координат:

$$\Delta x = X_2 - X_1, \Delta y = y_2 - y_1$$

Для определения номера квадранта, в котором содержится отрезок между центральными элементами меток  $M_x$  и  $M_y$ , используют знаки  $\Delta x$  и  $\Delta y$ .

Последовательность положений меток  $M_1$  и  $M_2$  во всех квадрантах и таблицу для определения номера квадранта, приведены на рис. 8.

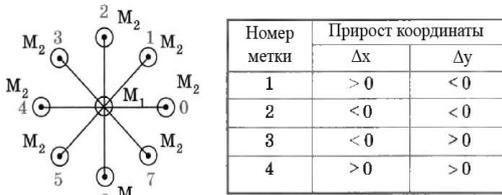


Рис. 8. Последовательность положений Меток  $M_1$  и  $M_2$

При адаптации алгоритмов распознавания к аффинным искажениям графических изображений удобно произвольный угол поворота графического изображения свести до угла его поворота в квадранте 1.

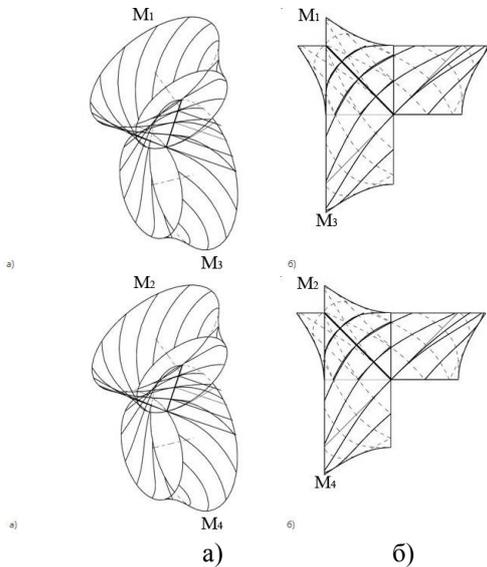


Рис. 9. Аффинные искажения модуля колесной пары: а) постоянство винтовой линии; б) совпадение направлений нормалей линий на делительном гиперboloиде и их со направленность линии контакта гиперboloидов

С этой целью необходимо вычислить константу квадранта (КК), (с учетом рис. 5) которая определяется так:

квадрант 1 -  $УК = 0$ ,

квадрант 2 -  $УК = 2$ ,  
 квадрант 3 -  $УК = 4$ ,  
 квадрант 4 -  $УК = 6$ .

Направление движения  $N$  в квадранте 1 связан с направлением  $N\phi$  фактического движения вдоль отрезка между центральными

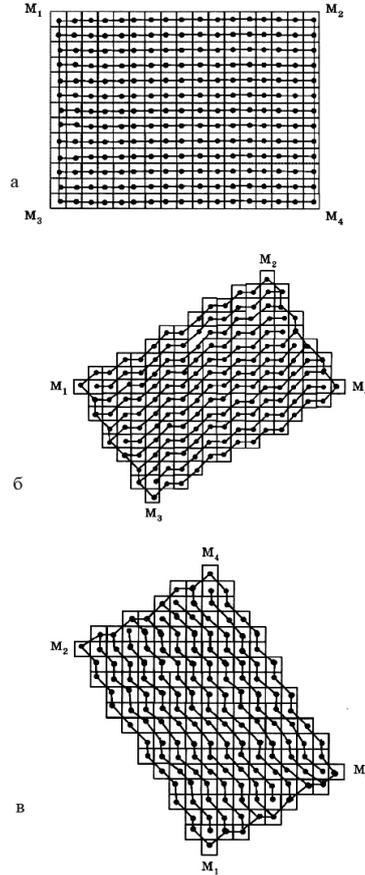


Рис. 10. Примеры построения трасс обхода элементов исходного (а) и искаженного графического изображения: б — при  $\phi < 45^\circ$ ; в — при  $\phi < 45^\circ$

элементами меток  $M_1$  и  $M_2$  и константой квадранта КК следующим соотношением:

$$N = N\phi - КК.$$

Например,  $N\phi = 3, КК = 2, N = 1; N\phi = 6, КК = 6, N = 0$ .

Сдвиг графического изображения определяется координатами  $x_1, y_1$  центрального элемента опорной метки  $M_1$ .

Масштаб  $M$  графического изображения определяется отношением выявленного значения длины диаметра метки  $D_v$  до его номинального значения  $D_n$ :

$$M = D_v / D_n < 1.$$

В процессе определения параметров аффинных искажений формируется ответ «Отказ от распознавания», если:

- количество обнаруженных меток меньше или больше от 4;

- длины диаметров обнаруженных меток выходят за допустимые пределы;
- коды диаметры обнаруженных меток не позволяют их идентифицировать;
- условия правильности расположения меток не выполняются.

Учет параметров аффинных искажений в алгоритмах распознавания графических

Любые алгоритмы распознавания изображений, в том числе и рукописных цифр, так или иначе связанные с обходом элементов

изображение то чаще всего как направления обхода используют горизонтальный (по строкам) и вертикальный (по столбцам).

Под влиянием аффинных искажений направления обхода элементов графического изображения и их.

Учитывая, что направления обхода элементов графического изображения искаженных по горизонталям и искаженных по вертикалям отличаются лишь константой квадрата  $KK = 2$ , для определения этих

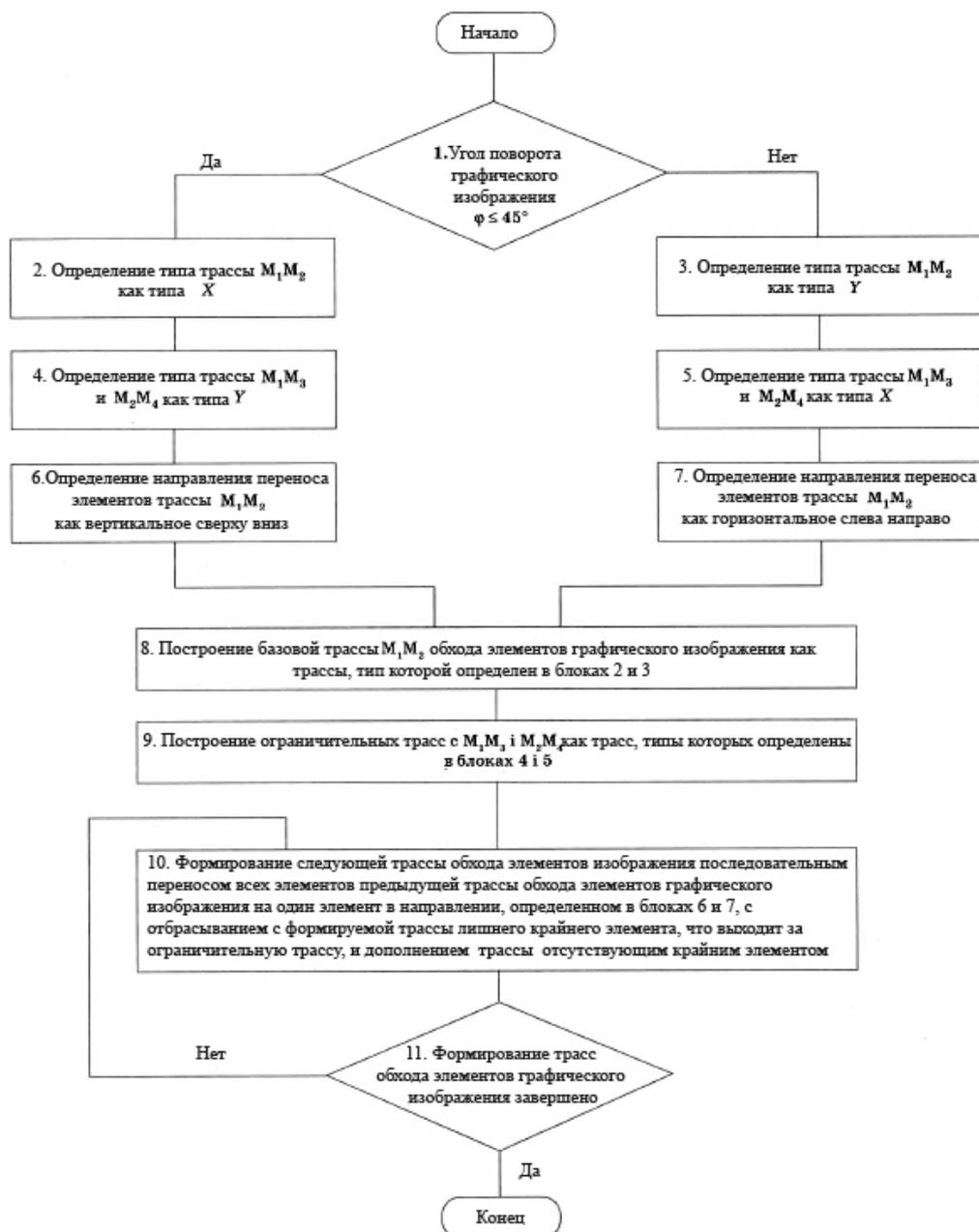


Рис. 11. Схема алгоритма построения трасс обхода элементов искаженного изображения

этих изображений. Поскольку в памяти ЭВМ подается матрица двухкоординатного

направлений можно использовать единый алгоритм.

Основная задача учета параметров аффинных искажений в алгоритмах распознавания графических изображений заключается в построении трасс обхода элементов графического изображения искаженных по горизонталям и искаженных по вертикалям, то есть в построении последовательностей направлений перехода между соседними элементами искаженного графического изображения, соответствующие направлениям обхода по горизонталям и по вертикалям первоначального графического изображения.

При этом необходимо обеспечить плотное прилегание соседних трасс такого обхода, то есть достичь такого взаимного расположения этих трасс, при котором ни один элемент рисунка не будет пропущен и ни один элемент не будет учитываться

Принцип построения трасс обхода элементов искаженного изображения, удовлетворяющие указанные требования плотного прилегания, наклонных линий на дискретной плоскости строят базовую трассу обхода элементов графического изображения, соединяющей центральные элементы меток  $M1$  и  $M2$ , и две ограничительные трассы, соединяющие центральные элементы меток  $Mx$  и  $M3$  и  $My$  и  $M4$ .

Соседняя трасса обхода элементов графического изображения строится параллельным переносом всех элементов

базовой трассы на один элемент в вертикальном направлении сверху вниз, если угол ( $\rho$  наклона линии  $M1M2$  удовлетворяет неравенство  $\varphi < 45^\circ$ , или на один элемент в горизонтальном направлении слева направо, если  $\varphi < 45^\circ$ .

Если при указанном переносе крайний элемент формируемой трассы хода элементов графического изображения выходит за ограничительную трассу, он отвергается, а если обнаруживается нехватка крайнего элемента – он дополняется в соответствии с рис. 2.

Примеры построения трасс обхода элементов графического изображения приведены на рис. 10.

Для построения трасс обхода элементов искаженного рисунка разработан специальный алгоритм, содержит 11 блоков (рис. 11)

Адаптация алгоритмов распознавания к аффинным графическим изображениям дает возможность решить важную задачу автоматизации производства модулей общих кругов, а именно гиперплоидных передач, и компенсации влияния указанных искажений за счет использования современных компьютерных технологий, а также выявить наличие проективных искажений графических изображений модулей общих кругов, исключив возможность их ошибочного распознавания.

## Affine transformations and coordinatization (axiomatics) for modules over common rings

Mzia Tskhomelidze

**Abstract.** The article addresses the issue of adapting recognition algorithms based on affine transformations and coordinatization for modules over common rings. Adaptation of recognition algorithms to affine graphic images makes it possible to solve the important problem of automating the production of modules of general circles, namely hyperploid transmissions, and compensating for the influence of these distortions through the use of modern computer technologies.

**Keywords:** affine transform recognition, graphic image, modules, concentric rings

### Литература

1. Э.А.И.Аиду. Сближение и наведение: Алгоритм среднеквадратичного приближения линий // Алгоритмы обработки экспериментальных данных М.: Наука, 1986. С. 23-30.
2. Э.А.И.Аиду. Оценивание параметров линии по случайно искаженным наблюдениям. // Обработка и представление данных в человеко-машинных системах. М.: Наука, 1988. С. 48-65.
3. D.B.Geselowitz. Dipole theory in electrocardiography. Am. J. Cardiol. 1964, 14:(9)301-6.
4. R.I.Jennrich. Asimptotic properties of nonlinear least squares estimation. The Annals of Math. Stat., 1969, т. 40, no, 2.