

**ОСНОВЫ  
ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ**

**ВЫЧИСЛИТЕЛЬНЫЕ  
НАНОСТРУКТУРЫ  
II  
ПРОГРАММНО-АППАРАТНЫЕ  
ПЛАТФОРМЫ**



**Основы информационных технологий**

# **ВЫЧИСЛИТЕЛЬНЫЕ НАНОСТРУКТУРЫ**

## **II Программно-аппаратные платформы**

Под редакцией д.т.н., профессора Г.М. Алакоза

**Учебное пособие**



Интернет-Университет  
Информационных Технологий  
[www.intuit.ru](http://www.intuit.ru)



БИНОМ.  
Лаборатория знаний  
[www.lbz.ru](http://www.lbz.ru)

Москва  
2010

# Оглавление

Введение .....	8
Лекция 1. Методы оценки вычислительных характеристик задач предметной области и поддерживающих их аппаратных платформ .....	23
1.1 Методика оценки требуемой пропускной способности вычислительных систем, ориентированных на решение «критических» задач предметной области .....	23
1.2 Особенности измерения основных характеристик вычислительных систем .....	30
1.3 Методы и средства оценки производительности вычислительных систем на основе стандартных инструментальных платформ .....	35
1.4 Методика определения алгоритмических затрат на решение «представительных» задач, включаемых в проблемно-ориентированную инструментальную платформу оценки пропускной способности (Б)ВС .....	43
Системотехнические выводы по лекции 1 .....	59
Лекция 2. Особенности использования программных инструментальных платформ параллельных вычислительных систем .....	61
2.1 Методы и средства представления и поддержки параллелизма .....	61
2.2 Средства поддержки параллелизма в системах программирования общего назначения .....	69
Системотехнические выводы по лекции 2 .....	88
Лекция 3. Специфика построения аппаратных платформ высокопроизводительных вычислительных систем с микропрограммным уровнем доступа .....	90
3.1 СБИС реализация .....	90
3.2 Особенности методики синтеза ассоциативно управляемых МКМД-бит-поточковых матричных СБИС арифметико-логической обработки данных .....	96
3.3 Синтез ассоциативно управляемых МКМД-бит-поточковых матричных СБИС арифметико-логической обработки данных по критерию максимума функциональной интеграции при минимуме аппаратных затрат .....	108
3.4 Синтез ассоциативно управляемых МКМД-бит-поточковых матричных СБИС арифметико-логической обработки данных по критерию максимума отказоустойчивости .....	130
Системотехнические выводы по лекции 3 .....	145

Лекция 4. МКМД-бит-поточковые субпроцессоры с (микро)программируемой архитектурой . . . . .	149
4.1. Ассоциативные конструкции на уровне слов-инструкций. . . . .	149
4.2. Базовая архитектура МКМД-бит-поточковых субпроцессоров и неоднородные двухуровневые вычислительные комплексы на их основе . . . . .	157
4.3. Ассоциативные конструкции в базовой системе команд МКМД-бит-поточкового (суб)процессора векторно-матричной обработки . . . . .	161
4.4. Методы и средства компенсации системных временных издержек в МКМД-бит-поточковых субпроцессорных трактах . . . . .	166
4.5. Порядок синтеза МКМД-бит-поточковых субпроцессорных трактов. . . . .	177
Системотехнические выводы по лекции 4 . . . . .	184
Лекция 5. Термальный синтез микропрограмм алгоритмически ориентированных МКМД-бит-поточковых субпроцессоров . . . . .	186
5.1. Технология (микро)программного конструирования МКМД-бит-поточковых субпроцессоров . . . . .	186
5.2. Термальный синтез DD-ассоциативных вычислительных конструкций алгоритмически ориентированного субпроцессора подстановки данных . . . . .	191
5.3. Термальный синтез алгоритмически ориентированного субпроцессора подстановки данных со встроенными PD-ассоциативными конструкциями . . . . .	202
Системотехнические выводы по лекции 5 . . . . .	214
Лекция 6. Инструментальные платформы МКМД-бит-поточковых вычислительных технологий . . . . .	217
6.1. Структура программных средств инструментальной платформы интерактивного микропрограммного конструирования . . . . .	217
6.2. Пользовательский интерфейс инструментальной платформы интерактивного микропрограммного конструирования . . . . .	221
6.3. Пользовательский интерфейс и операционное ядро инструментальной платформы термального синтеза микро программ векторно-матричных субпроцессоров . . . . .	233
6.4. Пользовательский интерфейс и операционное ядро инструментальной платформы обеспечения живучести МКМД-бит-поточковых субпроцессоров . . . . .	254
Системотехнические выводы по лекции 6 . . . . .	265

Лекция 7. Методы и средства обеспечения живучести и восстановления работоспособности МКМД-бит-поточковых субпроцессоров. . . . .	267
7.1. Методы и средства повышения отказоустойчивости МКМД-бит-поточковых субпроцессоров. . . . .	267
7.2. Системотехнические факторы, определяющие динамику парирования идентифицированных карт отказов. . . . .	273
7.3. Особенности рекуррентных процедур построения тестовых микропрограмм. . . . .	279
7.4. Особенности синтеза тест-данных и анализа откликов бит-процессоров . . . . .	287
7.5. Особенности термального синтеза тестовых микропрограмм в процессе эксплуатационной диагностики . . . . .	290
7.6. Оценка качества работы подсистемы локализации и идентификации отказов в МКМД-бит-процессорных матрицах . . . . .	304
Системотехнические выводы по лекции 7 . . . . .	312
Лекция 8. Оценка производительности и живучести МКМД-бит- потокового предпроцессора системы астронавигации . . . . .	315
8.1. Описание предметной области и системы преобразований МКМД-бит-поточкового субпроцессора слежения за «центром масс» астроориентира. . . . .	315
8.2. Поточковый алгоритм работы МКМД-бит-поточкового субпроцессора слежения за центром масс астроориентира . .	319
8.3. Система МКМД-бит-поточковых слов-инструкций поток-оператора слежения за центром масс астроориентира и их топология . . . . .	324
8.4. Анализ характеристик качества работы МКМД-бит-поточкового субпроцессора слежения за центром масс астроориентира . . . . .	354
Системотехнические выводы по лекции 8 . . . . .	366
Заключение. . . . .	358
Список использованных сокращений . . . . .	380
Литература . . . . .	383

## Введение

Междисциплинарный характер разработок средств вычислительной техники наиболее четко проявляется в проектах суперкомпьютеров всех поколений, требования к которым постоянно ужесточались. Последняя в этой области программа DARPA HPCS (high productivity computing systems) предусматривает, что перспективные суперкомпьютеры должны обеспечить реальную производительность в несколько петафлопс, а их глобально адресуемая оперативная память должна на 7–8 порядков превосходить существующую по интегральному показателю качества, из которой на пропускную способность приходится 4–5 порядков, а на емкость 3 порядка, которая должна быть доведена до нескольких петабайт. При этом ужесточаются и требования к технологиям разработки параллельных программ и к простоте поддерживающих их платформ.

Данная программа должна позволить США:

- устранить нарушенный паритет с Японией в области высокопроизводительных вычислений;
- создать суперкомпьютеры с перспективной архитектурой для решения стратегически важных государственных задач обеспечения национальной безопасности;
- продвинуть на военный и коммерческий рынок качественно новые технологии изготовления элементной базы и аппаратуры.

В современных условиях в России невозможно воспроизвести подобного рода глобальный междисциплинарный проект, требующий скоординированных усилий сотен проектных, научных и промышленных организаций и предприятий. Но у России не менее стратегически важные и масштабные государственные задачи обеспечения национальной безопасности, которые невозможно решить без создания программно-аппаратных платформ с сопоставимыми характеристиками.

Аналогичная ситуация складывалась в середине 80-х годов прошлого столетия в рамках программ СОИ и анти-СОИ, когда СССР вынужден был компенсировать технологическое отставание от США за счет гибких вычислительных технологий проектирования и работы алгоритмически ориентированных субпроцессоров с микропрограммным уровнем доступа, который обеспечивал (сверх)высокие по тем временам коэффициенты распараллеливания вычислений и повышал почти на порядок «время жизни» системы по сравнению с комплектами СБИС.

Полученный в этот период опыт раскрывается в этой книге, где вскрыты основные источники повышения производительности, отказоустойчивости и эффективного микропрограммного конструирования алгоритмически ориентированных субпроцессоров. В рамках этих про-

ектов пришлось отказаться от целого ряда традиционных подходов и оценок. В частности, пришлось перейти:

- от оценки производительности к расчету пропускной способности (сверх)многопроцессорной системы по потокам команд и данных;
- от традиционных системных платформ программирования к платформам микропрограммного конструирования, сходным с системными средствами кремниевой компиляции;
- от традиционных систем подстановки «горячего резерва» к системам перераспределения исполняемых (бит)инструкций согласно действующей в бит-матрице карте отказов.

В лекции 1 проанализированы апробированные на практике методы оценки вычислительных характеристик задач предметной области, которые служат базой разработки (Б)ВС, ориентированных на решение «критических» задач. Этот класс задач по своим требованиям превосходит не менее чем на порядок физические возможности существующей вычислительной техники, что вынуждает разработчиков (Б)ВС:

- задействовать в информационных технологиях «собственных нужд» комплекс интенсивных и экстенсивных факторов повышения производительности;
- снижать системные аппаратно-временные издержки на управление ходом вычислительного процесса, в том числе и за счет выбора или модификации вычислительных алгоритмов и их параметров.

Показано:

1. Метрологические проблемы вычислительной техники носят как фундаментальный, так и чисто методический характер, и они неразрешимы в принципе. Фундаментальная составляющая связана с отсутствием *единой метрики*, определяющей количество информации, а значит, и объем выполненной ЭВМ работы. Методическая составляющая связана с отсутствием эталонных ЭВМ, эталонных алгоритмов преобразования данных и стандартных процедур пересчета реальных алгоритмических затрат в затраты эталонного алгоритма. В итоге нарушается базовый принцип любого измерения, когда измерительный прибор является неотъемлемой частью измеряемого объекта (в данном случае операционной системы) и поэтому постоянно изменяет свою структурно-функциональную схему.
2. Все без исключения инженерные методики оценки производительности фактически отталкиваются от измерения времени, затраченного на решение «хорошо известной пользователю» задачи. В результате сопоставление двух ЭВМ весьма и весьма субъективно,

так как и замер времени, и оценка выполненных машиной инструкций, и распределение времени между системными и прикладными задачами носят достаточно субъективный характер.

3. Стандартные программные платформы оценки производительности и пропускной способности ЭВМ предоставляют пользователю только возможность создания тестовых программ, которые способны воспроизвести стабильную вычислительную нагрузку реальных задач только при использовании в них циклов *for*, так как при использовании циклов *if* вычислительная нагрузка зависит от содержимого преобразуемых данных.
4. Дальнейшее развитие программных платформ оценки производительности ЭВМ, скорее всего, будет связано с созданием адаптивных средств, оперативно реагирующих и на специфику алгоритма решения задачи, и на специфику архитектуры ЭВМ, и на содержимое преобразуемых потоков данных.
5. Снижение трудозатрат на адаптацию испытательных и исследовательских программ требует *структурного подхода*, в рамках которого программные модули представительных задач выступают в качестве неделимых единиц. Совокупность таких структурных единиц инструментальной платформы оценки качества ВС должна обеспечить управляемое перераспределение вычислительной нагрузки на операционные, коммутационные и управляющие ресурсы, а также ресурс памяти исследуемой ВС, для чего требуется точная априорная оценка этих же видов ресурсов, затребованных вычислительными алгоритмами представительных задач. Поэтому операционное ядро каждой представительной задачи должно быть алгоритмически прозрачным, чтобы на его основе можно было априорно и достоверно оценить требуемую нагрузку на все виды используемых ресурсов.

В лекции 2 раскрыты особенности построения и использования программных инструментальных платформ параллельных вычислительных систем общего назначения. Показано, что методы и средства *конструирования программ* являются не только прерогативой параллельных нейромикрокомпьютерных технологий и технологий с микропрограммным уровнем поддержки, но и широко используются при создании программных продуктов для ВС общего назначения. При этом надо отличать инструментальные платформы, которые ориентированы на использование уже существующих аппаратных платформ, от инструментальных платформ, ориентированных на их создание. В первом случае допускается только специфицированная реконфигурация аппаратной платформы под требования *вычислительного алгоритма* пользователя, что было апробиро-



вано в транспьютерных и многопроцессорных ЦПОС-проектах, а во втором случае вычислительный алгоритм известен и требуется методами и средствами (полу)казаного проектирования синтезировать аппаратуру его поддержки, что характерно для систолических матриц.

Центральная проблема создания таких инструментальных платформ — это *формализованные средства* представления и поддержки параллелизма, использование которых требует углубленного знания возможностей целевой аппаратной платформы, призванной воплотить в жизнь затребованный *программным конструктором* параллелизм. Решить в общем виде задачу формализованного представления параллелизма пока не удалось, что вынуждает:

- использовать *интерактивный* режим (микро)программного конструирования либо на всех стадиях проекта, как это имеет место в систолических и бит-поточковых технологиях, либо на отдельных, но самых ответственных этапах декомпозиции проекта и распределения (конфигурирования) затребованных ресурсов между коллективом вычислителей, как это имеет место в транспьютерных и многопроцессорных ЦПОС- и RISC-технологиях;
- инструментальные платформы «собственных нужд» строить по иерархическому принципу, поэтапно решая задачи извлечения из вычислительного алгоритма потенциально достижимого коэффициента распараллеливания вычислений, представления графа потока сигнала в «терминах» целевой аппаратной платформы, устранения синонимии, возникающей из-за ограниченных размеров целевой аппаратной платформы.

В лекции 3 раскрыты особенности работы одной из возможных аппаратных платформ ВС с микропрограммным уровнем доступа. За основу взята МКМД-бит-поточковая СБИС Н1841 ВФ1, которая была разработана в СССР в 1988 году и планировалась к использованию в широком спектре субпроцессорных трактов, ориентированных на решение «критических» задач аэрокосмических (Б)ВС двойного назначения. На ее основе показано:

1. Широко разрекламированные на Западе (возможно, и с подрывной целью) систолические вычислительные структуры ориентированы на ОКМД-режим распараллеливания вычислений и требуют интеллектуальных оболочек для приведения алгоритма пользователя к систолическому виду и кремниевых компиляторов для быстрого бездефектного проектирования специализированных систолических СБИС достаточно широкой номенклатуры, покрывающей потребности (Б)ВС во всем спектре решаемых задач. Поэтому в рамках систолических технологий преимущество получал не тот,

кто разрабатывал алгоритмы систолического типа, а тот, кто имел более эффективную СБИС-реализацию. (ОКМД — организация вычислений по типу «одиночный поток команд — множественный поток данных».)

2. Структурно-функциональный и схемотехнический синтез МКМД-бит-процессорных матричных СБИС, УБИС и систем на кремниевой пластине по сложности проекта и вытекающим из него требованиям к поддерживающим инструментальным программно-аппаратным платформам находится на уровне схем средней степени интеграции, содержащих не более 1000 логических вентилей. Это делает *прозрачным* процесс проектирования аппаратных платформ для таких вычислительных технологий как в структурно-функциональной, так и в диагностической плоскости. Такое кардинальное снижение размерности задач быстрого бездефектного проектирования и изготовления бит-матричных структур, давно уже не уступающих по уровню функциональной интеграции УБИС, приводит к существенному возрастанию сложности задач микропрограммного конструирования алгоритмически ориентированных субпроцессоров на их основе.
3. Различные варианты построения бит-процессорных матричных СБИС в основном разнятся средствами управления и коммутации, которые играют решающую роль в обеспечении отказоустойчивости СБИС и субпроцессоров на их основе. При этом зависящая от области применения система реализуемых бит-инструкций мало влияет на структурно-функциональную схему бит-процессора, которая в большей степени определяется принципами и методами организации вычислений.
4. Принципы и методы МКМД-бит-поточковой организации вычислений таковы, что их реализация приводит к естественной структурно-функциональной избыточности бит-процессоров, которая повышает отказоустойчивость аппаратных платформ. Во-первых, отказавший вентиль может оказаться неиспользуемым при реализации заданной бит-инструкции, а во-вторых, под воздействием отказа в бит-процессоре все же реализуется некоторая бит-инструкция, которая может принадлежать системе ее бит-операций и при сдвигах микропрограмм на бит-матрице может оказаться востребованной именно в данном месте бит-матрицы.
5. Одну из центральных проблем технологии прототипирования в рамках МКМД-бит-поточковых технологий можно решить на основе методов и средств *многоуровневого имитационного моделирования*, что позволяет уже на самых ранних этапах проектирования оценить влияние принимаемых конструктивных и технических решений на

отказоустойчивость создаваемого программно-аппаратного продукта. В результате удается сделать весь процесс проектирования полностью отечественных субпроцессорных трактов управляемым по фактору отказоустойчивости. При этом появляется достаточно уникальная возможность сбалансированного управления вводимой на этапе проектирования и используемой в процессе эксплуатации естественной структурно-функциональной и топологической избыточностью бит-процессоров и соответственно матриц на их основе. Это увеличивает степень связности проекта (суб)процессорного тракта в структурно-функциональной и диагностической плоскостях, так как с ростом структурно-функциональной избыточности возрастает и устойчивость к отказам бит-матриц, а значит, и накапливаемая в латентном периоде карта отказов, что негативно сказывается на эффективности работы подсистем диагностики и парирования карт отказов. Отсюда следует, что с ростом отказоустойчивости бит-матриц необходимо увеличивать частоту принудительного тестового контроля, что негативно сказывается на пропускной способности МКМД-бит-поточковых субпроцессорных трактов.

В лекции 4 раскрыты особенности микропрограммного конструирования МКМД-бит-поточковых субпроцессоров или, что одно и то же, особенности перехода с микрокомандного на ассемблерный уровень организации вычислений в (Б)ВС с микрокомандным уровнем доступа, которые сводятся к следующему.

1. Какие бы методы и средства распараллеливания вычислений ни применялись на различных уровнях управления ходом вычислительного процесса, рано или поздно архитектура (Б)ВС или ее отдельных компонент будет сведена к фон-неймановскому типу, который предполагает последовательный характер перечисления (микро) программных конструкций, описывающих либо бит-инструкцию, либо слов-инструкцию, либо поток-оператор, либо подпрограмму, либо программу, либо задачу, либо комплекс задач и т. д. В таких условиях МКМД-бит-поточковые вычислительные технологии удобны тем, что позволяют в едином операционном и аппаратном базисе оценить аппаратно-временные затраты на организацию вычислений на всех уровнях начиная со слов-командного.
2. Принципиальное отличие МКМД-бит-поточковых вычислительных технологий состоит в том, что обработка в них осуществляется в распределенной, **PD**-ассоциативной памяти (бит)инструкций, а не данных, как это имеет место в классических ассоциативных архитектурах, и не в сосредоточенных операционных устройствах

с фиксированной системой ассемблерных инструкций, как это имеет место в наиболее распространенных современных **RISC**-архитектурах.

3. Основным источником повышения пропускной способности МКМД-бит-поточковых субпроцессоров по потокам команд и данных является совмещение во времени и пространстве процессов передачи и обработки данных, что позволяет создавать структурно-функциональные схемы, адекватные графу связности операторов решаемой задачи, причем на всех уровнях организации вычислений начиная с бит-процессорного.
4. Технология *микропрограммного конструирования* МКМД-бит-поточковых субпроцессоров по объективным причинам носит итеративный и интерактивный характер. В таком процессе структурно-функциональная схема субпроцессора синтезируется декомпозицией задач пользователя, нисходящей до микропрограммных алгоритмов реализации отдельных операционных, управляющих, адресных и интерфейсных процедур, а оценку качества принятого технического решения можно получить на основе восходящего конструирования топологии всей микропрограммы, реализующей поток-оператор пользователя.
5. В процессе *микропрограммного конструирования* МКМД-бит-поточковых субпроцессоров можно применять практически все известные алгоритмические, структурно-функциональные, конструктивно-технологические и т. д. методы и средства компенсации системных временных издержек с той разницей, что в этом случае в качестве неделимой единицы проекта служит пространственно фиксированный поток бит-инструкций и пространственно-временной поток данных.
6. Несмотря на определенные неудачи в области оптоэлектронной вычислительной техники, исследования в этой области показали принципиальную возможность создания нетрадиционных вычислительных технологий, которые по максимуму используют функциональные возможности физических процессов, используемых в качестве вычислительных. В частности, в оптоэлектронных вычислителях можно отказаться от «священной коровы» традиционной вычислительной техники в виде регистровых схем управления, передачи и хранения информации и весь вычислительный процесс реализовать на основе *распределенной ассоциативной памяти данных*, связанной с произвольно коммутируемой вентиляционной матрицей *разветвленной системой связей прямого доступа*.

В лекции 5 показано, что проблемы интерактивного синтеза сверх- и гиперпараллельных микропрограммных конструкций можно преодолеть за счет *формализованного синтеза* библиотечных слов- и поток-инструкций на основе рекуррентных процедур, оперирующих не бит-процессорами, а *термами*, характерными для этих слов- и поток-инструкций. Методика термального синтеза фактически базируется на методах и средствах инструктированного синтеза сложных молекулярно-биологических соединений более простых. На первом шаге этой методики необходимо осуществить выбор между традиционными **DD**-ассоциативными и перспективными **PD**-ассоциативными конструкциями или, что одно и то же, между методами адаптации, который зависит от «времени жизни» гетероструктуры:

- если «время жизни» велико, то предпочтение следует отдавать **DD**-ассоциативным конструкциям, конформационные преобразования которых позволяют настроить их на реализацию требуемой функции;
- если «время жизни» мало, то предпочтение следует отдавать **PD**-ассоциативным конструкциям, настройка которых на реализуемую функцию и содержимое одного из преобразуемых операндов требует структурной адаптации.

В рамках этой методики интерактивный характер выбора топологии формально сохраняется только для термов, но фактически он предопределен линейными размерами матриц, априори выделяемых на реализацию топологии слов- и поток-инструкций. При этом, как и в молекулярной биологии, следует отличать *функционально значимые и промежуточные* (вставочные) *термы*, первые из которых задают правила и циклы взаимодействия между слов-инструкциями, а вторые обеспечивают рекуррентное наращивание по какому-либо параметру однородной функции. Характерно также, что в «активных зонах» **DD**- и **PD**-ассоциативные конструкции взаимодействуют не только между собой, но и с потоками управляющих данных, которые задают циклы такого взаимодействия.

Показательно, что в МКМД-бит-поточковых матрицах «фронт вычислительной волны» распространяется по бит-матрице с помощью механизмов, сходных «пробоям» в реальных аксонно-коллатеральных связях, что наиболее четко проявляется в интерфейсах. Это указывает на то, что в реальных нейросетях преобразование информации может осуществляться не только в соме нейрона, но и в процессе распространения возбуждения по аксонно-коллатеральным связям, реализующим функции алгоритмически ориентированных **PD**-ассоциативных субпроцессоров.

В лекции 6 раскрыты особенности построения и работы реально существующих и не имеющих зарубежных аналогов программных инструментальных платформ бит-поточковых информационных технологий «собственных нужд». Как и в кремниевой компиляции, в бит-поточковых технологиях необходимо поддержать проект в трех плоскостях: структурно-функциональной, программно-аппаратной и диагностической, а само развитие проекта должно осуществляется по иерархии от выбора физико-технического процесса, закладываемого в основу бит-матричной СБИС и до задач пользователя (сверх)многопроцессорной (Б)ВС. В результате:

1. Эффективность объектно-ориентированного подхода к созданию инструментальных платформ МКМД-бит-поточковых технологий предопределяют два фактора: отношение иерархии между «объектами» проекта и множественность «вложений» одних и тех же компонент в программно-аппаратную и диагностическую плоскости проекта.
2. В структурно-функциональной и программно-аппаратной плоскости бит-процессор является неделимой единицей проекта, но при переходе в диагностическую плоскость его необходимо раскрыть, включив в иерархию описания блоки, узлы, вентили и физико-технические процессы, нарушение условий правильной работы которых и служит источником «расширения» библиотеки бит-инструкций бит-процессора.
3. Интерактивный режим, а с ним и графический интерфейс инструментальных платформ пока остаются атрибутом бит-поточковых технологий, так как они поддерживают завершающий этап проектирования субпроцессоров. На этом этапе решается узловая задача всего проекта, связанная с компоновкой библиотечных модулей на бит-матрице, или, что одно и то же, на плоской ортогональной **FIFO**-регистрающей коммутационной решетке, где сдвиг на один дискрет всегда сопровождается изменением времени задержки на 1 такт. (**FIFO** — «первый вошел — первый вышел».)
4. Главная особенность МКМД-бит-поточковых технологий состоит в том, что объектом диагностики на схемотехническом уровне является относительно простая схема бит-процессора (порядка 800–1000 вентиляей), которая к тому же достаточно просто разбирается на блоки с разным последствием отказов: регистр команды, операционный канал, канал транзита, средства коммутации и т. п. Вместе с тем функциональные возможности программно-аппаратной плоскости проекта, как правило, отвечают ультравысокой степени интеграции (порядка 8–20 миллионов вентиляей и

выше) за счет наращивания «площади» бит-матрицы (от 1000 до 10000 бит-процессоров).

5. Средства *эксплуатационной диагностики* МКМД-бит-поточковых субпроцессоров являются продолжением средств *промышленной диагностики*, используемых для генерации эталонных тестов и эталонных тестовых воздействий для первой. Незнание производственных методов и средств обнаружения отказов в бит-матричных СБИС и положенных в их основу физических моделей отказов приводит к неоправданному увеличению пространства диагностики, делая само изделие диагностонепригодным. Данный вывод справедлив для всех изделий микроэлектроники, и поэтому стратегия их заимствования третьих стран чревата высокой вероятностью необнаружения отказов, особенно в современных процессорах **RISC**- и **ЦПОС**-архитектуры, которые по степени функциональной интеграции уже давно относятся к УБИС.

В *лекции 7* раскрыты специфические особенности обеспечения живучести и восстановления работоспособности (фактически микропрограммного «ремонта» без отключения источников питания) МКМД-бит-поточковых субпроцессоров, которые вытекают из возможности использования некратного (по отношению ко всей бит-матрице) резерва. Приведенные данные позволяют утверждать:

1. В МКМД-бит-поточковых вычислительных технологиях на основе микроэлектронных СБИС или УБИС основная ставка сделана на структурно-функциональный *полиморфизм* бит-процессоров, который поддерживает как технологию восходящего микропрограммного конструирования поточковых (суб)процессоров, так и технологию достаточно быстрого парирования карт отказов за счет *параметрической адаптации* исходного «рабочего тела» микропрограммы пользователя. При этом используется как скрытый, почти 2-кратный структурно-функциональный резерв каждого бит-процессора, так и централизованный и существенно меньший по отношению к рабочей области бит-матрицы «горячий» аппаратный резерв, дифференцированно «перемещаемый» по всей бит-матрице.
2. Наибольших интеллектуальных и временных затрат требует этап *интерактивного микропрограммного конструирования* проблемно- или алгоритмически ориентированных (суб)процессоров, который завершается созданием индивидуальной вторичной топологической структуры для каждого поток-оператора пользователя. При переходе вычислительной техники в супрамолекулярную или нанометровую область этот этап *трансформируется в синтез* «рабочего

тела», для которого *специфицированы* (био)химические модальности и уровни «сигналов», кодирующих управляющие, тестовые и обрабатываемые потоки информации.

3. Этап *синтеза* тестовых микропрограмм достаточно просто *формализовать* и свести к размещению ограниченного количества термов на бит-матрице с учетом ограничений на конструкцию гальванических программных шин и шин данных (**P**- и **D**-шины соответственно), обеспечивающих доступ к периферийным бит-процессорам.
4. Основная проблема, препятствующая кардинальному снижению времени локализации и идентификации отказов в бит-матрице, связана с опосредованным доступом к «внутренним» бит-процессорам по **FIFO**-регистровым **P**- и **D**-шинам, и решать ее приходится в основном *конструктивно-технологическими методами*, напрямую зависящими от используемых физико-химических процессов управления, передачи и обработки информации. Сам процесс диагностики приобретает *корректирующий характер*, по крайней мере на отдельных фазах и этапах.
5. Удовлетворить темп реального времени при решении задач локализации и идентификации отказов можно только с использованием комплекса системотехнических, схемотехнических и конструктивно-технологических мер, которые минимизируют временные издержки на программирование бит-матрицы, снижают размеры контролируемых подматриц и повышают эффективность парирования обнаруженных карт отказов непосредственно во время диагностики.
6. Парирование карт отказов сводится к индивидуальной, толерантной переконпоновке вторичной топологической структуры каждого поток-оператора пользователя. При этом локальные и глобальные *аффинные преобразования* «рабочего тела» микропрограммы по максимуму сохраняют условия пространственно-временного взаимодействия бит-инструкций. Но это требует трансформации части «функциональных» бит-процессоров в «коммутационные» и пространственного «сдвига» бит-инструкций по бит-матрице.
7. Выбор системы локальных, толерантных, аффинных преобразований представляет собой достаточно сложную, двухкритериальную оптимизационную задачу, которая в супрамолекулярных вычислениях может оказаться соизмеримой по временным и энергетическим затратам с повторным синтезом (структурной регенерацией) исходного «рабочего тела», в процессе которого нерелевантные молекулы замещаются релевантными. Это указывает на то, что в супрамолекулярных технологиях прежде всего необходимо осуществить выбор между ОКМД-схемой парирования отказов, которая



основана на прямой подстановке резерва (*структурная адаптация*), и МКМД-схемой, которая основана на *параметрической адаптации* структурно-функциональной схемы вычислителя.

В лекции 8 синтезирован МКМД-бит-поточковый субпроцессор, ориентированный на решение задач астронавигации космических летательных аппаратов, и на его основе проиллюстрированы особенности использования информационных технологий «собственных нужд» с использованием микропрограммного уровня доступа. На этой основе показано:

1. В МКМД-бит-поточковых вычислительных технологиях структурно-функциональная декомпозиция заданий пользователя представляет собой многокритериальную задачу, в которой необходимо учесть:
  - минимум аппаратных затрат и максимум неоднородности составляющих модулей, повышающий устойчивость к отказам,
  - минимум информационных связей между модулями, снижающий затраты на редактирование этих связей после проведения над телом микропрограммы толерантных аффинных преобразований,
  - и целый ряд других требований к топологии микропрограммы, обеспечивающих удобство работы всей системы обеспечения живучести субпроцессора.
2. Естественная структурно-функциональная избыточность МКМД-бит-поточковых субпроцессоров используется комплексно, обеспечивая не только (сверх)высокие коэффициенты распараллеливания вычислений, но и повышая более чем на порядок их отказоустойчивость в сравнении с параллельными системами на основе процессоров традиционной архитектуры, а также вычислительную устойчивость, гарантируя нулевую абсолютную погрешность за счет вариации разрядной сетки по ходу вычислений.
3. Чем выше МКМД-уровень распараллеливания вычислений, тем большему количеству процессоров необходимо задать управляющие параметры и начальные условия, что в пределе, когда вычисления организованы по принципу «одна (бит)инструкция — один (бит)процессор» приводит к работе всей (Б)ВС в режиме интерпретации, а не компиляции программ.
4. Работа МКМД-бит-поточкового субпроцессора *в режиме интерпретации*, а не трансляции или компиляции программ:
  - сопряжена с временными издержками, которые сгруппированы в отдельный этап ввода микропрограммы в бит-матрицу и которые окупаются при выполнении системообразующего неравенства за счет большого количества циклов обработки «больших» потоков данных;

- требует приведения исходного вычислительного алгоритма к потоковому виду с явным указанием всех управляющих, интерфейсных и диагностических процедур;
  - приводит к росту в основном аппаратных, а не временных затрат на реализацию индивидуальных управляющих, интерфейсных и диагностических процедур, что повышает «мобильность» модуля при парировании карт отказов.
5. Чем больше имеющийся естественный структурно-функциональный и топологический резерв у бит-процессоров и соответственно матриц на их основе, тем более устойчив к отказам поток-оператор и тем больше карта отказов, накапливаемая в латентном периоде. Отсюда следует, что с ростом отказоустойчивости бит-матриц необходимо увеличивать частоту принудительного тестового контроля, что негативно сказывается на пропускной способности МКМД-бит-поточковых субпроцессорных трактов.
  6. С проектных позиций МКМД-бит-поточковые вычислительные технологии удобны тем, что на самых ранних этапах создания субпроцессоров здесь можно получить достоверные оценки достижимой пропускной способности по потокам инструкций и данных, массо-габаритам, потребляемой энергии, отказоустойчивости и времени задержки, которые играют решающую роль в оценке летно-технических и боевых характеристик космических ЛА и их БЭО.
  7. Методы и средства обеспечения живучести МКМД-бит-поточковых субпроцессоров не исключают, а только дополняют традиционные методы и средства, основанные на введении дополнительной аппаратной, информационной и временной избыточности. При этом возможность проведения в темпе реального времени фактически ремонтно-восстановительных работ позволяет снизить необходимый аппаратный резерв до десятков процентов и сохранить живучесть субпроцессорного тракта при наличии в нем карт, насчитывающих десятки отказов. В результате удельные аппаратные затраты на парирование одного отказа в МКМД-бит-поточковых субпроцессорах не менее чем на порядок меньше аналогичных затрат в традиционных фон-неймановских схемах мажоритарного резервирования, которые по-прежнему доминируют в зарубежной авионике.
  8. Даже на отсталой по современным меркам элементной базе разработки середины 80-х годов прошлого столетия в МКМД-бит-поточковых технологиях удается достичь производительности не менее  $3,6 \cdot 10^3$  MIPS и пропускной способности по потокам данных

порядка 500 Мбит/сек, что более характерно для современных персональных ЭВМ, работающих на частотах порядка единиц ГГц.

Авторы отдают себе отчет в том, что с целым рядом базовых положений могут не согласиться глубокие и авторитетные специалисты из более узких областей научной деятельности. Но такова цена междисциплинарного подхода к решению сложных проблем, в рамках которого трудно договориться даже по терминологии и тем более по «незыблемым истинам», закладываемым в систематику научно-технических исследований и разработок. Поэтому авторы руководствовались не бесспорностью базовых положений и предложенной системы ценностей при создании сложных технотронных комплексов, а полнотой охвата всей проблематики их сквозного системного проектирования, в рамках которого нет и не может быть научных и организационно-технических «мелочей», пренебрежение которыми, как правило, и приводит к провалу всего проекта.

Разделы 3.4, 6.4 написаны А.А. Поповым, а разделы 7.5 и 8.4 совместно А.А. Поповым и Г.М. Алакозом. Разделы 1.2 и 1.3 написаны А.П. Сериковым, а раздел 2.2 — М.В. Кураком. Остальной материал книги написан Г.М. Алакозом.

В отработку технологии микропрограммного конструирования и в создание инструментальных платформ МКМД-бит-поточковых технологий кроме авторов учебного пособия внесли вклад П. Авдошкин, А. Костин и А. Добротворский. При этом за основу была взята более ранняя разработка НИИ Космического приборостроения, решающий вклад в которую внес М.А. Глазков.

Аппаратная платформа МКМД-бит-поточковых технологий в виде СБИС Н1841 ВФ1 реализована на производственном объединении «Альфа» (г. Рига, 1988 г.) большим коллективом разработчиков, среди которых в первую очередь необходимо отметить Л. Вольперта, И. Михайлова, И. Луцкого, В. Горохова, Г. Страутманиса, А. Лейниекса, П. Малакса, Т. Барютину, О. Кузьмина и целый ряд инженеров, обеспечивших комплекс измерительных и испытательных мероприятий при подготовке и выпуске этой СБИС.

Эффективность аффинных преобразований в МКМД-бит-поточковых технологиях исследована совместно Г.М. Алакозом, А.П. Сериковым и В.В. Ткачевым. «Нечисленные» модели формальных нейронов разработаны и исследованы Г.М. Алакозом совместно с И. Клейменовым и А.А. Саломатовым.

После распада СССР исследования технологий микропрограммного конструирования были поддержаны командованием Военно-воздушной инженерной академии им. проф. Н.Е. Жуковского, и в пер-

вую очередь генерал-полковником авиации, дважды Героем Советского Союза, летчиком-космонавтом В.В. Коваленком, генерал-лейтенантом В.П. Кутаховым, генерал-лейтенантом В.А. Ефимовым, генерал-майором Ю.Б. Кулифеевым, полковником В.П. Харьковым и коллективом кафедры электронной автоматики.

Всем перечисленным коллегам, а также корректору А. Ангиной, редактору С. Перепелкиной и руководителю издательского отдела Н. Рахмановой авторы выражают глубокую благодарность и признательность.

# Лекция 1. Методы оценки вычислительных характеристик задач предметной области и поддерживающих их аппаратных платформ

## 1.1. Методика оценки требуемой пропускной способности вычислительных систем, ориентированных на решение «критических» задач предметной области

Реальная ценность вычислительных систем (ВС), ориентированных на решение «критических» задач предметной области, определяется качеством оценки требуемой пропускной способности по потокам инструкций и данных. При этом разработчику ВС известны следующие исходные данные, характеризующие предметную область:

- алгоритм или совокупность возможных алгоритмов преобразования информации;
- объемы, моменты поступления и точность представления входных данных;
- точность, интервалы времени или циклы получения результатов вычислений.

Заказчика или покупателя таких ВС в первую очередь интересуют аппаратно-временные затраты на реализацию его алгоритмов, а системные издержки организации вычислений им воспринимаются как накладные расходы. Поэтому *алгоритмические* затраты должны быть удовлетворены в полном объеме, так как именно они определяют потребительскую ценность создаваемой ВС, а *системные* аппаратно-временные издержки необходимо минимизировать в ходе разработки, чтобы добиться приемлемого соотношения цены и качества.

В отличие от ЭВМ общего назначения, к которым можно отнести и персональные компьютеры, предметная область создаваемых (Б)ВС всегда характеризуется рекордными параметрами, но при этом она достаточно четко очерчена и потенциальному пользователю хорошо известны представленные выше характеристики. Это обеспечивает высокую степень достоверности оценки *требуемой* пропускной способности по потокам команд и данных, диктуемых алгоритмом и продолжительностью цикла решения задачи, зная которые, уже можно выбрать необходимую архитектуру и требуемый коэффициент распараллеливания вычислений.

При таком подходе к проектированию или выбору (Б)ВС:

- главной целью является поддержание темпа реального времени, диктуемого предметной областью;
- главным ограничением является частота  $F_c$  или время цикла  $\tau_c = 1/F_c$  исполнения инструкции, достижимые в рамках освоенных или осваиваемых технологий производства аппаратуры;
- варьируемыми параметрами при поиске компромисса в соотношении цены и качества являются «сложность» алгоритмов и вытекающие из нее точность и объемы преобразуемой информации.

Такая методика оценки качества (Б)ВС исходит из *безусловной поддержки темпа реального времени*, диктуемого предметной областью, и для его реализации прежде всего необходимо получить аналитическую оценку «длины» программы  $L_p$  *вычислительного* алгоритма «критической» задачи, которую в традиционных вычислительных технологиях измеряют максимальным количеством *ассемблерных инструкций*, реально расходуемых на ее решение. В нетрадиционных вычислительных технологиях «длину» программы приходится измерять в тех (микро)программных единицах, для которых известен или задан цикл исполнения.

Основная трудность оценки «длины» программы  $L_p$  связана с наличием в *вычислительном алгоритме* операторов условных переходов типа *if*, которые ставят ее в прямую зависимость от *содержимого* обрабатываемых потоков данных, которые в данном случае необходимо рассматривать как *случайные*. Для (Б)ВС вносимая операторами *if* «случайность» и вытекающая из нее неопределенность в длине программы ослабляется тем обстоятельством, что в высокоответственные применения оператор условных переходов должен *мажорироваться* оператором *for*, который, с одной стороны, призван лимитировать максимально допустимую по темпу реального времени продолжительность исполнения каждого цикла программы, а с другой стороны, должен повысить ее устойчивость к информационным сбоям, разрывая «бесконечные» циклы, чреватые потерей управления опасными технотронными комплексами двойного назначения.

Зная «длину» программы  $L_p$ , можно выбрать тип архитектуры, «идеальной» в том смысле, что при этом были учтены только алгоритмические затраты, а системные издержки считались нулевыми. Для этого достаточно воспользоваться соотношениями (1.5)–(1.7) раздела 1.5 [307] и проверить неравенства:  $L_p < \max L(\text{ОКОД}) = \Delta T_k / \tau_c$ ,  $L_p < \max L(\text{ОКМД}) = T_k / \tau_c$  или  $L_p < \max L(\text{МКМД}) = T_0 / \tau_c$ , где  $\Delta T_k$  — дискрет времени, диктуемый теоремой Котельникова,  $T_k = \Delta T_k * N$  — цикл формирования циклически обрабатываемой выборки данных из  $N$  операндов, а  $T_0$  — допустимое время задержки в (Б)ВС. Если ни одно из этих неравенств не выполняется, то необходимо переходить на уровень распараллеливания отдельных ассемблерных инструкций, то есть к архитектурам с микрокомандным

уровнем доступа, гарантирующим максимально возможное функциональное насыщение каждого такта работы (Б)ВС. Как показывает практика, выйти за рамки «идеальной» архитектуры разработчику в дальнейшем не удастся, и ему остается только разрабатывать комплекс мер, направленных на минимизацию системных аппаратно-временных издержек.

Данная методика апробирована при создании спецпроцессоров *обработки сигналов и изображений*, как на основе заказных СБИС, так и на основе систолических и бит-матричных СБИС [70, 138–144]. Указанные аппаратные платформы поддерживают предметную область, для которой характерны [70]:

1. *Обработка точек изображения*, которая представляет собой преобразование полутонного изображения, выравнивание гистограмм, переквантование, отображение яркости и т. п.
2. *Фильтрация сигналов и изображений*, которая может осуществляться сравнением с эталоном, оконными методами, сверткой и/или корреляцией, линейной фазовой, медианной, винеровской, калмановской, адаптивной и т. п. фильтрацией.
3. *Матричная алгебра*, которая включает сингулярное (однозначное) разложение, геометрическое вращение, оценки максимальной энтропии или правдоподобия, вычисление псевдообращения и восстановление изображения, стохастическое оценивание параметров и т. п.
4. *Преобразования сигналов и изображений*, к которым в первую очередь относят преобразования Фурье, Хаара, Хафа, Адамара, Карунена — Лозва и т. п.
5. *Сортировка*, которую можно выполнить методом «слияния», «пузырька», и т. п.

Признав, что циклы *for* являются неотъемлемой составляющей таких (Б)ВС, решаемые ими задачи можно разбить на классы с характерной зависимостью количества арифметико-логических операций от количества операндов  $N$  в *циклически обрабатываемых* потоках данных (таблица 1.1 [70]).

Численные значения таблицы 1.1 соответствуют параметрам «критических» задач, актуальных на конец 80-х годов прошлого столетия. Тем не менее уже тогда требовались (Б)ВС с производительностью порядка  $10^{12}$ – $10^{15}$  ассемблерных операций в секунду. Если принять во внимание, что цикл исполнения ЦПОС- и RISC-инструкций был уже постоянен и в тот период времени не превышал 1 ГГц, становится ясно, что суммарный коэффициент распараллеливания вычислений должен был находиться на уровне  $10^4$ – $10^6$  и был достижим только в архитектурах МКМД-типа с микрокомандным уровнем доступа.

Однако (Б)ВС решают не одну и не одни только «критические» задачи. Поэтому их операционный ресурс постоянно перераспределяется по

**Таблица 1.1.** Требуемая производительность алгоритмов цифровой обработки сигналов и изображений

Тип алгоритма	Необходимая производительность
<b>Линейные, типа <math>O(N)</math>:</b>	
Пространственная фильтрация	$10^2$ – $10^5$ MOPS
Свертка	
Обнаружение контуров	
<b>Второго порядка, типа <math>O(N^2)</math>:</b>	
Сортировки	$10^3$ – $10^7$ MOPS
Медианная фильтрация	
Классификация ближайшего соседа	
<b>Третьего порядка, типа <math>O(N^3)</math>:</b>	
Векторно-матричная алгебра	$10^4$ – $10^8$ MOPS
Спектральные вычисления	
Адаптивная фильтрация	

ходу вычислений между решаемыми задачами, что требует учета системных издержек на организацию вычислений. Данное обстоятельство только подтверждает тот факт, что в любой нетрадиционной архитектуре (Б) ВС всегда найдется уровень управления фон-неймановского типа. В простейшем случае такое управление выполняется по жесткой циклограмме и связанные с ним издержки в основном сводятся к плановой обработке «прерываний», которые неминуемо вызывают «паразитные» перемещения программ и данных по иерархии запоминающих устройств (Б)ВС: регистры — КЭШ — ОЗУ — внешние ЗУ и обратно.

Но циклограмма жестко регламентирует только порядок (пере)запуска задач, время решения которых может флуктуировать в достаточно больших пределах (1,5–3 раза и более) из-за наличия циклов по условию. Объясняется это тем, что с помощью мажорирующих циклов *for* удается ограничить только *максимально допустимое* время выполнения этого цикла, а реальные временные затраты могут оказаться значительно меньше, если сработает оператор *if*. В результате трудно рассчитать как время обработки одного планового «прерывания», которое зависит от фазы выполнения прерванной задачи, так и количество таких «прерываний». Основным источником такой неопределенности являются паразитные перезагрузки памяти на участках кеш — ОЗУ — внешние ЗУ, что вынуждает брать в кавычки термин «прерывание».

Отсюда, даже в этом простейшем режиме управления системные временные издержки носят *случайный характер с очень высокой дисперсией* моментов запуска каждой последующей фазы вычислительного процесса.



[ . . . ]