

## Лекции

<b>Лекция 1.</b> Предпосылки, вызовы, модели мира современных высокопроизводительных вычислений . . . . .	11
<b>Лекция 2.</b> Классификация современных приложений и многоядерных систем . . . . .	23
<b>Лекция 3.</b> Высокопроизводительные многоядерные процессоры для встраиваемых приложений . . . . .	29
<b>Лекция 4.</b> Многоядерные процессоры с низким энергопотреблением . . . . .	45
<b>Лекция 5.</b> Графический процессор G80 . . . . .	63
<b>Лекция 6.</b> Проект Larrabee . . . . .	72
<b>Лекция 7.</b> Проект ATAC . . . . .	79
<b>Лекция 8.</b> Мультиядерные процессоры ARM-архитектуры . . . . .	91
<b>Лекция 9.</b> Мультиядерные процессоры на базе MIPS-совместимых ядер . . . . .	102
<b>Лекция 10.</b> Мультипроцессор Cell . . . . .	109
<b>Лекция 11.</b> Процессоры AMD . . . . .	117
<b>Лекция 12.</b> Процессоры архитектуры SPARC . . . . .	131
<b>Лекция 13.</b> Процессоры Intel . . . . .	139
<b>Лекция 14.</b> Состояние отечественных разработок в области высокопроизводительных вычислений . . . . .	163
<b>Лекция 15.</b> Сравнительные характеристики процессоров . . . . .	189
<b>Лекция 16.</b> Приемы и технологии программирования многоядерных процессоров . . . . .	197
<b>Лекция 17.</b> Средства программирования многоядерных процессоров . . . . .	213

# Содержание

Введение .....	9
<b>Лекция 1.</b> Предпосылки, вызовы, модели мира современных высокопроизводительных вычислений .....	11
Предпосылки .....	11
Проблемы .....	11
Мнения ведущих специалистов .....	13
Возможные пути обхода проблем .....	18
Краткие итоги .....	21
<b>Лекция 2.</b> Классификация современных приложений и многоядерных систем .....	23
Сферы применения многоядерных процессоров .....	23
Основные классы многоядерных процессоров (many-core, multi-core, MIMD, SIMD, SMP, MMP) .....	26
Краткие итоги .....	27
<b>Лекция 3.</b> Высокопроизводительные многоядерные процессоры для встраиваемых приложений .....	29
«Ячеистые» процессоры компании Tiler .....	29
Процессор CSX700 .....	39
Краткие итоги .....	44
<b>Лекция 4.</b> Многоядерные процессоры с низким энергопотреблением .....	45
167-ядерная вычислительная платформа — AsAP-II .....	45
Процессоры SEAforth .....	52
Краткие итоги .....	61
<b>Лекция 5.</b> Графический процессор G80 .....	63
Структура графического процессора G80 .....	63
Модель программирования CUDA .....	68
Краткие итоги .....	70
<b>Лекция 6.</b> Проект Larrabee .....	72
Краткие итоги .....	77
<b>Лекция 7.</b> Проект ATAC .....	79
Предпосылки .....	79
Основные идеи архитектуры ATAC .....	80

Ключевые элементы технологии АТАС .....	82
Структура межъядерных связей .....	85
Передача данных и согласование кэш-памяти .....	87
Краткие итоги .....	89
<b>Лекция 8. Мультиядерные процессоры ARM-архитектуры .....</b>	<b>91</b>
Процессоры с IP-ядрами ARM .....	92
Общая характеристика мультиядерных ARM-процессоров .....	92
ARM11 MPCore .....	94
Серия Cortex-A MPCore .....	95
ARM Cortex-A9 MPCore .....	97
ARM Cortex-A5 .....	99
Краткие итоги .....	101
<b>Лекция 9. Мультиядерные процессоры на базе     MIPS-совместимых ядер .....</b>	<b>102</b>
Когерентная процессорная система MIPS32 1004К .....	102
Система на кристалле XLP832 .....	104
Краткие итоги .....	108
<b>Лекция 10. Мультипроцессор Cell .....</b>	<b>109</b>
Общая структура процессора Cell .....	109
Структура процессорного элемента Power (PPE) .....	110
Структура SPE (Synergistic Processor Elements) — «синергичного» процессорного элемента .....	112
Краткие итоги .....	116
<b>Лекция 11. Процессоры AMD .....</b>	<b>117</b>
Хронология процессоров AMD .....	117
Мультиядерные AMD Obteron .....	121
Краткие итоги .....	129
<b>Лекция 12. Процессоры архитектуры SPARC .....</b>	<b>131</b>
Хронология современных процессоров SPARC .....	131
Ядро SPARC .....	134
Краткие итоги .....	138
<b>Лекция 13. Процессоры Intel .....</b>	<b>139</b>
Хронология процессоров Intel .....	139
Краткие итоги .....	161

<b>Лекция 14.</b> Состояние отечественных разработок в области высокопроизводительных вычислений . . . . .	163
Микропроцессор 1891ВМЗ . . . . .	166
Микросхемы сигнальных процессоров серии «Мультикор» . . . . .	168
Микросхема сигнального процессора 1892ВМЗТ . . . . .	173
Микросхема сигнального процессора 1892ВМ2Я . . . . .	177
Микросхемы сигнальных процессоров 1892ВМ5Я (МС-0226, ЦПОС-02) и 1892ВМ4Я . . . . .	181
Телекоммуникационный микропроцессор NVCom-01 . . . . .	185
Краткие итоги . . . . .	188
<b>Лекция 15.</b> Сравнительные характеристики процессоров . . . . .	189
Интегральные характеристики . . . . .	189
Производительность отдельных ядер . . . . .	191
Краткие итоги . . . . .	195
<b>Лекция 16.</b> Приемы и технологии программирования многоядерных процессоров . . . . .	197
Проблемы параллельного программирования . . . . .	197
Модель выгрузки функций (function offload model) . . . . .	202
Модель ускорения вычислений (computational acceleration) . . . . .	203
Потоковые модели (streaming models) . . . . .	204
Модель мультипроцессора с разделяемой памятью (shared-memory multiprocessor model) . . . . .	207
Модель ассиметричных потоков (asymmetric thread runtime model) . . . . .	207
Краткие итоги . . . . .	211
<b>Лекция 17.</b> Средства программирования многоядерных процессоров . . . . .	213
Специализированные библиотеки . . . . .	213
Языки программирования . . . . .	216
Краткие итоги . . . . .	219
Заключение . . . . .	221
Общий глоссарий . . . . .	222
Список сокращений . . . . .	232
Предметный указатель . . . . .	238
Литература . . . . .	241

## Введение

В данном курсе предпринята попытка создания общей картины развития многоядерных процессоров. Рассмотрены структуры процессоров различных классов, структура вычислительных ядер и связей между ними. Рассмотрены как процессоры, обозначаемые в иностранной литературе термином «*many-core processor*», под которым понимается процессор, имеющий десятки или сотни вычислительных ядер в одном кристалле, так и мультиядерные процессоры — «*multi-core processors*», имеющие относительно небольшое количество ядер на кристалле.

Основной целью является стремление показать области применения и сформировать у будущих специалистов теоретические знания и практические навыки по принципам организации современных многоядерных процессоров и программных средств для решения широкого спектра задач в различных областях, а именно: познакомить с основами построения высокопроизводительных систем; привить навыки программирования многоядерных процессоров с использованием различных техник.

Основные задачи курса:

- дать представление о современных тенденциях развития микропроцессоров;
- дать представление об основных архитектурах многоядерных процессоров и систем на их основе;
- развить умение ставить задачи и выбирать инструментальные средства при работе с многоядерными вычислительными платформами.

Курс состоит из нескольких разделов.

Лекции 1-2 затрагивают общие вопросы, связанные с тенденциями и постулатами развития процессов. Представлены мнения ведущих специалистов на проблемы внедрения и эксплуатации многоядерных процессоров, дается краткая характеристика современных распространенных задач и приложений.

В лекциях 3-4 приводится описание многоядерных процессоров, обладающих количеством ядер от нескольких десятков и выше.

Лекции 5-6 посвящены графическим процессорам.

Несколько особняком стоит седьмая лекция — она посвящена одному из экспериментальных проектов, связанных с применением оптических линий для связи процессорных ядер на кристалле.

Лекции 8-9 описывают мультиядерные процессоры (с небольшим числом ядер) на базе IP-ядер архитектур ARM и MIPS.

В лекциях с 10 по 13 представлены высокопроизводительные мультиядерные процессоры, наиболее часто встречающиеся в практике, — это процессоры фирм IBM, AMD, Sun, Intel.

Лекция 14 представляет отечественные мультиядерные процессоры ряда серий.

В лекции 15 обобщаются различные характеристики процессоров, представленных в лекциях 3-14 для формирования общей картины.

Лекции 16-17 затрагивают некоторые проблемы и вопросы программирования многоядерных процессоров, а также общие проблемы, связанные с переходом на многоядерные системы.

## Лекция 1. Предпосылки, вызовы, модели мира современных высокопроизводительных вычислений

Лекция посвящена основным проблемам развития средств вычислительной техники и причинам, обуславливающим современный подход к построению микропроцессорных систем.

**Цель лекции:** показать актуальность перехода к параллельной обработке информации и данных и, в частности, к многоядерным процессорам. Дать представление об основных подходах к увеличению производительности систем.

### Предпосылки

По мере развития технических средств изменения окружающего мира, меняются потребности человека к производительности вычислительных систем, меняются и требования приложений. Одним из возможных решений является разработка и применение систем с массовым параллелизмом — **кластеров, Grid-систем**, многопроцессорных комплексов, систем на многоядерных процессорах.

Компьютерный мир все плотнее и плотнее объединяется с миром реальным. Данный процесс основывается на успехах полупроводниковой промышленности, расширении и удешевлении сетевых услуг, развитии беспроводных технологий связи. Появился специальный термин «все-проникающие компьютерные системы» — pervasive computing. Компьютер перестал быть ЭВМ, электронной вычислительной машиной, и неожиданно выяснилось, что в своем первоначальном виде компьютер не вполне соответствует окружающей среде. Реальный мир во всех его проявлениях параллелен, а современный компьютерный мир по своей природе последователен: данные передаются по последовательным каналам, команды выполняются одна за другой, и, как следствие, — любые попытки распараллеливания и адаптации к условиям реального мира рождают чрезвычайно сложные и искусственные решения [1, 2].

### Проблемы

Одну из фундаментальных слабостей современных компьютеров вскрыл в 1977 году Джон Бэкус в речи, с которой он выступил при получении Тьюринговской премии. С тех пор в ход пошло прозвучавшее в ней выражение «бутылочное горло фон Неймана», символизирующее собой органический недостаток архитектурной схемы, которая ассоциируется с

именем Джона фон Неймана. Альтернативной архитектурой является «**гарвардская**» архитектура, называемая так по имени машины Говарда Айкена Harvard Mark I, где программы хранились на перфорированных лентах, а обрабатываемые данные — в релейных массивах. В современных микроконтроллерах перфоленты заменила память ROM, реле — память RAM, но суть осталась прежней: в процессоре встречаются два независимых и не сдерживающих друг друга потока команд и данных.

Происходит смена приоритетов при разработке процессоров, систем на их основе и, соответственно, программного обеспечения. Направление смены приоритетов выражает мнение [1] Дэйва Паттерсона, профессора Калифорнийского университета в Беркли, изобретателя **RISC-процессоров** и **RAID-массивов**, который считает, что грядет новое время, когда перестанут действовать прежние прописные истины и им на смену придут новые. Однако переход этот осложняется наличием трех «стен».

- **Энергетическая стена.** Старая истина — энергия не стоит ничего, транзисторы дороги. Новая истина — дорога энергия, транзисторы не стоят ничего.
- **Стена памяти.** Старая истина — память работает быстро, а операции с плавающей запятой медленны. Новая истина — системную производительность сдерживает память, операции выполняются быстро.
- **Стена параллелизма на уровне команд.** Старая истина — производительность можно повысить за счет качества компиляторов и таких архитектурных усовершенствований, как конвейеры, внеочередное выполнение команд, сверхдлинное командное слово (Very Long Instruction Word, VLIW), явный параллелизм команд (Explicitly Parallel Instruction Computing, EPIC) и др. Новая истина — естественный параллелизм, команды допустимы и длинные, и короткие, но выполняются они параллельно, на разных ядрах.

Выделяют также так называемую «стену ментальности» (educational wall). За время существования компьютеров сложилась определенная образовательная традиция. Вообще говоря, несовершенство и избыточная затратность современных компьютерных систем становится популярной темой для обсуждения. По мнению Билла Инмона, одного из создателей хранилищ данных, все современные решения для работы с данными объединяет то, что стоимость функциональных компонентов (транзисторы, диски и т.д.) становится все менее значимой, а основная стоимость продуктов приходится на поддерживающую инфраструктуру.

Помимо очевидных технологических претензий к **фон-неймановской архитектуре** (излишняя сложность, неспособность к распараллеливанию, низкий коэффициент полезного использования транзисторов и т.п.) есть еще одна группа претензий. Несколько десятилетий назад существовал



повышенный интерес к искусственному интеллекту, делались прогнозы на возможности компьютеров с системами искусственного интеллекта. Ныне же на то, чтобы приспособить компьютеры для решения интеллектуальных задач, остались, в основном, языковые возможности. Многие прогнозы не оправдались.

За последнее десятилетие удельная производительность процессоров в пересчете на число транзисторов упала на один-два порядка. Дальнейшее развитие полупроводниковых технологий по **закону Мура** не может компенсировать неэффективность современных процессоров. Диспропорцию между числом транзисторов на кристалле и удельной производительностью, которая выражена в скорости выполнения, отнесенной к площади, иногда называют «дефицитом Мура» (Moore Gap). Показатели быстродействия процессоров (в частности, тактовые частоты) достигли практически граничных показателей, плотность энергии увеличивается пропорционально уменьшению размеров транзисторов, и, соответственно, увеличиваются проблемы с теплоотводом.

Нетрудно прийти к выводу, что если нельзя использовать возможности закона Мура на одном ядре из-за исключительной сложности такого ядра, то следует пойти по пути увеличения числа ядер. Именно так поступили в Sun Microsystems, выпустив 8-ядерный процессор Niagara. Количественный рост, постулируемый законом Мура, предопределяет неизбежные перемены, и в какой-то момент количество должно перейти в качество. Сторонники EPC-подхода также склонились к многоядерному решению. На форуме IDF осенью 2004 года Пол Отеллини, генеральный директор Intel, заявил: *«Мы связываем наше будущее с многоядерными продуктами; мы верим, что это ключевая точка перегиба для всей индустрии... Это не гонка. Это радикальное изменение в компьютеринге, мы не имеем права рассматривать происходящее как простую технологическую гонку».*

## Мнения ведущих специалистов

Настрой разработчиков процессоров можно оценить по следующим их словам [1].

Ананд Агарвал, профессор Массачусетского технологического института, основатель компании Tera, выступая на конференции, посвященной проблемам современных процессоров, сказал следующее: *«Процессоры становятся все более и более единообразными и обезличенными, критичнее становится система».* *«Процессор — это транзистор современности».*

Энди Бехтольсхайм (Sun Microsystems): *«Больше нельзя рассматривать производительность как серьезное ограничение. Теперь главное ограничение — ввод-вывод, а производительность ввода-вывода невозможно поднять за счет простого увеличения плотности транзисторов».*

Журнал Dr. Dobb's Journal, август 2006 года: *«Только подумайте: впервые в истории вычислительной техники массовые компьютеры не будут строиться на основе фон-неймановской архитектуры — они станут параллельными. Наши приложения смогут одновременно выполняться на нескольких ядрах. Такой переход сулит огромные возможности, но и потребует много усилий. Конечно, принцип параллельных вычислений не является каким-либо новшеством. Их исследовали многие из специалистов, упомянутых ранее. Такие компании, как Cray, занимаются параллелизмом уже много лет. Но большинство программистов, создающих приложения для массовых вычислительных систем, не использовали эту технологию повседневно. Только сейчас начинается процесс переноса принципов многопоточности и параллелизма в область массового программирования».*

Можно говорить о двух заметно различающихся между собой тенденциях в процессе увеличения числа ядер. Одна носит название **мультиядерность** (multi-core). В этом случае предполагается, что ядра являются высокопроизводительными и их относительно немного; сейчас их число — два-четыре, и согласно закону Мура оно будет периодически удваиваться. Основных недостатков два: первый — высокое энергопотребление, второй — высокая сложность чипа и, как следствие, низкий процент выхода готовой продукции. При производстве 8-ядерного процессора IBM Cell только 20% производимых кристаллов являются годными. Другой путь — **многоядерность** (many-core). В таком случае на кристалле собирается на порядок большее число ядер, но имеющих более простую структуру и потребляющих милливатты мощности. Сейчас количество ядер варьируется от 40 до 200, и если закон Мура будет выполняться и для них, то можно ожидать появления процессоров с тысячами и десятками тысяч ядер. Данный подход попадает в категорию так называемых «разрушительных инноваций» (disruptive innovation). Ни одна из современных школ программирования не в состоянии справиться с грядущими проблемами. Неслучайно на суперкомпьютерной конференции, прошедшей в Дрездене в 2007 году, один из центральных докладов назывался «Развал системы традиционных знаний в многоядерную эру. Все что вы знаете — неверно».

Очевидно, что увеличение в десятки и сотни раз числа все тех же фон-неймановских ядер на одной подложке не является панацеей. В одном из интервью Дональд Кнут сказал по поводу упрощенного подхода к увеличению числа ядер следующее: *«Мне кажется, у проектировщиков процессоров иссякли идеи, и они хотят переложить ответственность за невозможность повышать производительность компьютеров в соответствии с законом Мура на тех, кто создает программы. Они предлагают процессоры, отлично работающие на отдельных тестах, но я не удивлюсь, что вся эта эпопея многоядерности закончится неменьшим провалом, чем Itanium,*

*где все выглядело прекрасным, пока не выяснилось, что компиляторы с соответствующими ему возможностями предвидения невозможно написать.*

*Сколько вы знаете программистов, относящихся с энтузиазмом к будущим многоядерным процессорам? У всех, кого я знаю, они лишь вызывают огорчение, хотя разработчики процессоров говорят, что я неправ. Я знаю приложения, адаптируемые к параллельному исполнению; это графический рендеринг, сканирование изображений, моделирование биологических и физических процессов, но все они требуют подходов, которые чрезвычайно специализированы. Моих знаний хватило, чтобы написать о них в «Искусстве программирования», но я считаю время, затраченное на них, потерянным, в этой области все быстро меняется, и совсем скоро написанное мной никому не будет нужным. Мультиядерность в том виде, как ее представляют, сейчас не прибавляет мне счастья».*

Многоядерные процессоры, если все сводится к размещению большего числа классических простых ядер на одной подложке, нельзя воспринимать как решение всех проблем. Их чрезвычайно сложно программировать, они могут быть эффективны только на приложениях, обладающих естественной многопоточностью.

На конференции Hot Chips, которая прошла в Стэнфордском университете в конце августа 2009 года, производители рассказали о своих планах по выпуску восьмиядерных серверных процессоров [4].

Традиционным средством повышения производительности процессоров всегда было увеличение тактовой частоты, но рост потребления энергии и проблемы с отводом тепла заставили разработчиков вместо этого наращивать количество ядер.

Представители IBM впервые подробно рассказали об архитектуре Power7, которая приходит на смену представленной два года назад Power6, предназначенной для Unix-серверов корпорации. Архитектура Power7 является для IBM заметным шагом вперед. От двухъядерных моделей корпорация переходит к четырем, шести и восьми ядрам, и каждое из них способно выполнять четыре потока команд одновременно.

Процессоры Power7 должны выйти в первой половине 2010 года. Они будут изготавливаться по 45-нанометровому технологическому процессу и, как утверждают в IBM, их можно будет устанавливать в нынешних серверах моделей Power 570 и Power 595.

Инженеры Fujitsu рассказали о восьмиядерном процессоре SPARC64 — новой версии выпущенного в июле прошлого года четырехъядерного процессора SPARC64 VII. Об этом процессоре, получившем кодовое наименование Venus, вкратце упоминалось в докладе на конференции Hot Chips 2008 года, но никаких подробностей не приводилось. Процессоры SPARC64 используются в серверах Fujitsu и Sun Microsystems (правда, планы последней в свете слияния с Oracle повисли в воздухе).

Представители AMD сделали доклад о применении в лезвийных серверах процессоров Magny-Cours. Это 12-ядерные процессоры, которые объединяют на одной микросхеме два шестиядерных процессора, соединенных шиной AMD Hyper Transport. Название им взяли у известной гоночной трассы во Франции. Выход Magny-Cours запланирован на начало 2010.

Специалисты Intel сообщили о работе над восьмиядерными двухточными процессорами Nehalem-EX, выход которых назначен на первую половину 2010 года. Но о Tukwila — четырехъядерной версии процессора Itanium — в их докладе сказано не было. Его появление уже несколько раз откладывалось и теперь ожидается в будущем году.

Но вопрос в том, в какой степени современное программное обеспечение — и, следовательно, конечные пользователи — способны использовать возможности многоядерных процессоров. Приложения должны быть написаны так, чтобы решаемые ими задачи можно было разбивать на подзадачи, выполняемые на нескольких ядрах параллельно.

Аналитик компании Gartner Карл Кланч в январе подготовил пессимистический доклад на эту тему. Производители процессоров, пишет он, вывели количество ядер далеко за тот уровень, для которого были разработаны основные программные средства — операционные системы, инструментарий промежуточного слоя, приложения, виртуализационные средства и т. д.

Большая часть современных программных систем, считает Кланч, обладает ограничениями, которые не дадут им воспользоваться всеми имеющимися ядрами. Пользователям же, чтобы угнаться за развитием аппаратной платформы, придется чаще обновлять программное обеспечение.

Аналогичное мнение выразили в июне аналитики IDC. За некоторыми исключениями, считают они, производительность большинства современных приложений при наличии свыше четырех логических процессоров стагнирует.

Другие смотрят на вещи более оптимистично. Большая часть программ, для которых важна производительность процессора, полагают они, на сегодняшний день уже параллелизована, потому что их разрабатывали в расчете на кластеры серверов или на крупные системы SMP-архитектуры со множеством процессоров.

*«Не думаю, что это станет препятствием, и любой, кто так думает, вероятно, слишком пессимистичен»*, — считает аналитик Натан Бруквуд из компании Insight64 [4].

Одной из наиболее интересных точек зрения на происхождение «стен» и других недостатков существующего компьютерного миропорядка обладает немецкий профессор Рейнер Хартенштайн, с 1977 года возглавляющий лабораторию Xputer Lab, одну из немногих, где занима-

лись тематикой, связанной с реконфигурируемым компьютерингом (Reconfigurable Computing).

Профессор Хартенштайн — автор таких терминов, как «антимашина» (anti-machine), «конфигурируемое аппаратное обеспечение» (configware), «структурное проектирование аппаратного обеспечения» (structured hardware design) и ряда других. Хартенштайн является создателем KARL, новаторского языка для проектирования аппаратного обеспечения. Реконфигурирование — это перепрограммирование аппаратуры; теоретически его можно применять к массивам, собранным из ядер с фон-неймановской архитектурой, но гораздо перспективнее — к ядрам, представляющим собой антимашину. Рамамути в ряде своих работ показал, что чаще всего компьютеры с массовым параллелизмом, содержащие тысячи и десятки тысяч процессоров, оказываются менее продуктивными, чем ожидается при их проектировании. Основные причины этого эффекта, называемого самим Рамамути «суперкомпьютерным кризисом», — наличие «стены памяти» и сложность программирования параллельных задач, что тоже является следствием фон-неймановской архитектуры. Главный вывод Хартенштайна состоит в том, что и энергетическая стена, и стена памяти, и стена параллелизма на уровне команд являются прямым следствием фон-неймановской архитектуры. Из-за технологических ограничений Джоном фон Нейманом была избрана схема, в основе которой лежит управляемый поток команд, программа, выбирающая необходимые для команд данные. Этим определяется канонический состав архитектурных компонентов, составляющих любой компьютер, — счетчик команд, код операции и адреса операндов; он остается неизменным по сей день. Все мыслимые и немыслимые усовершенствования архитектуры фон Неймана в конечном счете сводятся к повышению качества управления потоком команд, методам адресации данных и команд, кэшированию памяти и т.п. При этом последовательная архитектура не меняется, но сложность ее возрастает. Очевидно, что представление о компьютере как об устройстве, выполняющем заданную последовательность команд, лучше всего подходит для тех случаев, когда объем обрабатываемых данных невелик, а данные являются статическими. Но в современных условиях приходится сталкиваться с приложениями, где относительно небольшое количество команд обрабатывает потоки данных. В таком случае целесообразно предположить, что компьютером может быть и устройство, которое имеет каким-то образом зашитые в него алгоритмы и способно обрабатывать потоки данных. Такие компьютеры могли бы обладать естественным параллелизмом, а их программирование свелось бы к распределению функций между большим числом ядер.

## Возможные пути обхода проблем

Можно допустить существование следующих альтернативных схем. Одна, фон-неймановская, предполагает, что вычислительным процессом управляет поток команд, а данные, в основном статичные, выбираются из каких-то систем хранения или из памяти. Вторая схема основывается на том, что процессом вычислений управляют входные потоки данных, которые на входе системы попадают в подготовленную вычислительную инфраструктуру, обладающую естественным параллелизмом. С точки зрения реализации первая схема гораздо проще, кроме того, она универсальна, программы компилируются и записываются в память, а вторая требует специальной сборки нужной для определенной задачи аппаратной конфигурации. Вторая схема старше; пример тому — табуляторы, изобретенные Германом Холлеритом и с успехом использовавшиеся на протяжении нескольких десятилетий. Корпорация IBM достигла своего могущества и стала одной из самых влиятельных компаний в США, производя электромеханические табуляторы для обработки больших массивов информации, не требующей выполнения логических операций. Их программирование осуществлялось посредством коммутации на пульте, а далее устройство управления в соответствии с заданной программой координировало работу остальных устройств.

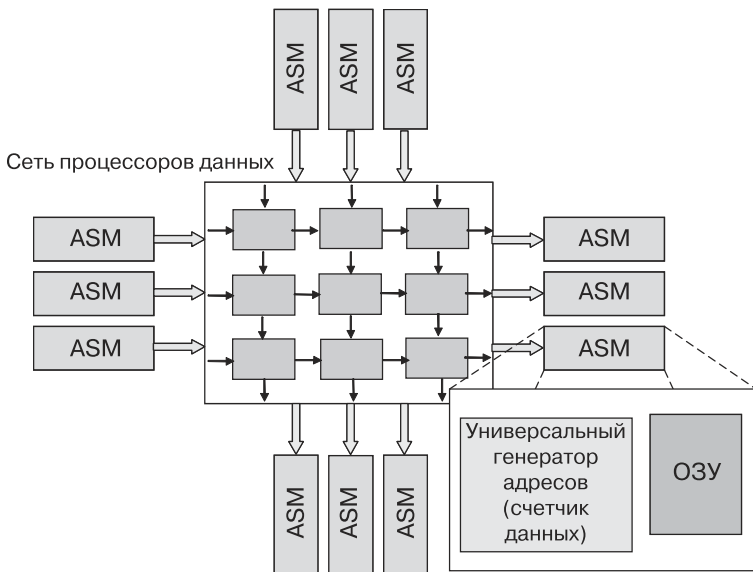


Рис. 1.1. Антимашина