МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

А.Д. Чередов, А.Н. Мальчуков

ОРГАНИЗАЦИЯ ЭВМ И СИСТЕМ

*Рекомендовано в качестве учебного пособия
Редакционно-издательским советом
Томского политехнического университета*

4-е издание, переработанное и дополненное

Издательство
Томского политехнического университета
2016

УДК 004.38+004.7(075.8)
ББК 32.973+32.973.202я73
Ч–462

Чередов А.Д., Мальчуков А.Н.

Ч–462 Организация ЭВМ и систем: учебное пособие / А.Д. Чередов, А.Н. Мальчуков; Томский политехнический университет. – 4-е изд., перераб. и доп. – Томск: Изд-во Томского политехнического университета, 2016. – 236 с.

В учебном пособии рассматриваются основные вопросы, связанные с организацией ЭВМ и систем: архитектуры, характеристики и классификация ЭВМ; функциональная и структурная организация ЭВМ и центрального процессора; принципы организации подсистемы памяти ЭВМ и вычислительных систем; принципы организации подсистемы ввода-вывода; архитектуры и способы организации многопроцессорных вычислительных систем.

Учебное пособие подготовлено на кафедре вычислительной техники ТПУ и предназначено для студентов ИДО, обучающихся по направлению 09.03.01 «Информатика и вычислительная техника».

УДК 004.38+004.7(075.8)
ББК 32.973+32.973.202я73

Рецензенты

Кандидат технических наук,
зам. начальник управления информационных технологий
ОАО «Востокгазпром»
П.М. Острасть

Кандидат технических наук,
доцент кафедры программирования ТГУ
С.А. Останин

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

В последнее десятилетие в России бурно осуществляется информатизация и компьютеризация всех сфер человеческой деятельности. Компьютеры или электронные вычислительные машины (ЭВМ), оснащенные специальным программным обеспечением, являются технической базой и инструментом для вычислительных, информационных и автоматизированных систем.

При изучении дисциплины «Организация ЭВМ» в процессе подготовки бакалавров по направлению «Информатика и вычислительная техника» студенты используют знания, умения и навыки, полученные по дисциплинам «Информатика», «Дискретная математика», «Основы теории передачи информации», «Введение в информационные технологии», которые являются пререквизитами данной дисциплины.

Целью дисциплины «Организация ЭВМ» является освоение теоретических основ функциональной и структурной организации ЭВМ, включающих архитектуры ЭВМ, технические характеристики, классификации, особенности организации различных типов ЭВМ и ее составных частей (процессора, памяти, ввода-вывода), а также современное состояние и тенденции развития средств вычислительной техники.

В результате изучения этой дисциплины студент должен уметь применять полученные знания для решения практических задач: проводить анализ всего многообразия типов ЭВМ, осуществлять анализ параметров основных технических средств ЭВМ, выбирать, комплексировать и тестировать аппаратные средства вычислительных систем. Он должен владеть навыками конфигурирования компьютеров различного назначения.

Знания, умения и навыки, полученные при изучении дисциплины «Организация ЭВМ», необходимы для освоения ряда других дисциплин: «Операционные системы», «Периферийные устройства», «Сети и телекоммуникации» и др.

В процессе освоения дисциплины у студентов формируются компетенции, дающие им возможность разрабатывать технические задания на оснащение отделов, лабораторий, офисов компьютерным оборудованием, осуществлять наладку и обслуживание этого оборудования.

В предлагаемом вниманию читателя учебном пособии рассматриваются современные проблемы, связанные с функциональной и структурной организацией ЭВМ и её составных частей. Учебное пособие состоит из шести глав.

В первой главе даются основные понятия и определения, относящиеся к ЭВМ, подробно рассматриваются архитектуры компьютеров (SISD, SIMD, CISC, RISC, VLIW, EPIC, MISD, MIMD), приводятся технические и эксплуатационные характеристики, классификации ЭВМ с кратким описанием истории развития, современного состояния и функциональных особенностей различных типов компьютеров (мэйнфреймов, суперЭВМ, рабочих станций, серверов, персональных компьютеров и т.д.).

Вторая глава посвящена изложению основ функциональной и структурной организации ЭВМ. В ней содержится описание типов данных, используемых в процессорах интеловской архитектуры (IA-32, IA-64), способов адресации данных, структур и форматов команд CISC- и RISC-процессоров, наборы расширения системы команд x86 (MMX, SSE, SSE2, SSE3, SSE4 и др.), обобщенной структуры ЭВМ.

В третьей главе излагаются основы функциональной и структурной организации центрального процессора ЭВМ: определяется состав и назначение основных устройств процессора; описываются регистровые структуры процессоров IA-32, x86-64, IA-64, особенности многоядерных микроархитектур Intel Core, Intel Nehalem, Intel Sandy Bridge, Intel Haswell, Intel Skylake; структуры универсальных микропроцессоров Intel, AMD. Особенности организации процессоров семейства «Эльбрус».

В четвертой главе описываются принципы организации подсистемы памяти компьютера: рассматривается иерархическая структура памяти компьютера; способы организации кэш-памяти; принципы организации оперативной памяти и методы повышения её пропускной способности. Особое внимание уделяется реализации виртуальной памяти.

В пятой главе рассматриваются особенности организации системного интерфейса и ввода/вывода информации: даётся общая характеристика и классификация интерфейсов; описываются способы организации передачи данных и системной организации компьютеров на базе чипсетов Intel и AMD.

В шестой главе кратко описываются архитектуры и классификации многопроцессорных и многомашинных вычислительных систем (MISD, MIMD, SMP, MPP и др.).

В приложении 1 приведены контрольные вопросы и задания для самопроверки.

1. АРХИТЕКТУРЫ, ХАРАКТЕРИСТИКИ, КЛАССИФИКАЦИЯ ЭВМ

Электронная вычислительная машина (компьютер) – комплекс технических и программных средств, предназначенных для автоматической обработки информации в процессе решения вычислительных и информационных задач.

Под **системой** понимают любой объект, который одновременно рассматривается и как единое целое, и как объединенная в интересах достижения поставленных целей совокупность разнородных элементов.

Вычислительная система – взаимосвязанная совокупность средств вычислительной техники, включающая не менее двух основных процессоров либо вычислительных машин. Основным процессором называют составную часть ЭВМ, которая выполняет вычисления, предусматриваемые алгоритмами решаемых задач.

Информационная система – взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели. Информационная система немыслима без персонала, взаимодействующего с компьютерами и телекоммуникациями.

Под **архитектурой** ЭВМ понимается общая функциональная и структурная организация машины, определяющая методы кодирования данных, состав, назначение, принципы взаимодействия технических средств и программного обеспечения.

Можно выделить следующие важные для пользователя компоненты архитектуры (рис. 1.1):

а) функциональные и логические возможности процессора (система команд, форматы команд и данных, способы адресации, разрядность обрабатываемых слов и т.д.);

б) структурную организацию и принципы управления аппаратными средствами (центральным процессором, памятью, вводом/выводом, системным интерфейсом и т.д.);

в) программное обеспечение (операционная система, трансляторы языков программирования, прикладное программное обеспечение и т.д.).

1.1. Архитектуры ЭВМ

Исторически первыми появились однопроцессорные архитектуры. Классическим примером однопроцессорной архитектуры является архитектура фон Неймана со строго последовательным выполнением ко-

манд: процессор по очереди выбирает команды программы и также по очереди обрабатывает данные. По мере развития вычислительной техники архитектура фон Неймана обогатилась сначала конвейером команд (рис. 1.2), а затем многофункциональной обработкой и по классификации М. Флина получила обобщенное название SISD (Single Instruction Single Data – один поток команд, один поток данных).

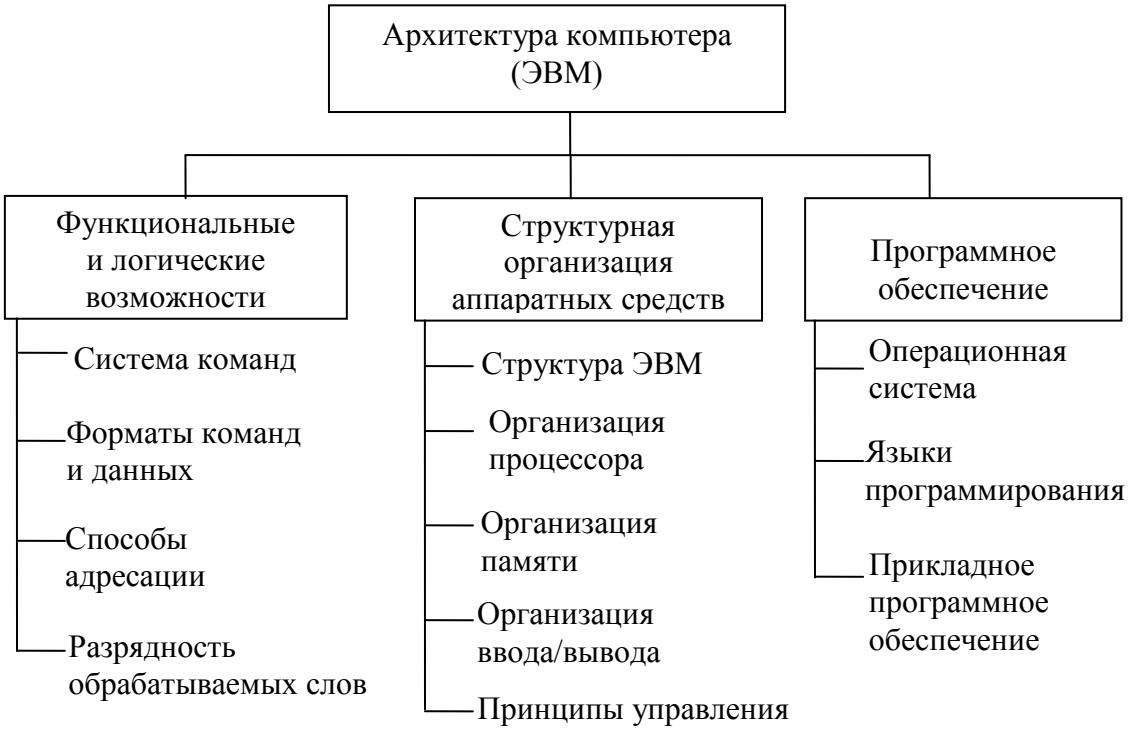


Рис. 1.1. Основные компоненты архитектуры ЭВМ

Архитектуры класса SISD охватывают те уровни программного параллелизма, которые связаны с одинарным потоком данных.

Параллелизм циклов и итераций тесно связан с понятием множественности потоков данных и реализуется векторной обработкой. В классификации компьютерных архитектур М. Флина выделена специальная группа однопроцессорных систем с параллельной обработкой потоков данных – SIMD (Single Instruction Multiple Data, один поток команд – множество потоков данных).

Ведущие поставщики микропроцессоров ищут пути повышения их производительности и снижения энергопотребления за счет использования многоядерных структур процессоров и многопотоковой обработки команд.

По классификации М. Флина такие структуры можно отнести к архитектурам MISD (Multiple Instruction Single Data, множество потоков команд – один поток данных) и MIMD (Multiple Instruction Multiple Data, множество потоков команд – множество потоков данных).

Рис. 1.2. Развитие и классификация однопроцессорных архитектур

1.1.1. Конвейерная обработка команд

Процедура выполнения команд процессором включает несколько характерных этапов. В простейшем случае можно выделить, как минимум, четыре этапа обработки команд (рис. 1.3, *а*): выборка команды (ВК), декодирование команды (ДК), выполнение операции (ОП) и запись результата (ЗР).

Каждый этап в процессоре выполняется за один такт. При последовательной обработке команд (рис. 1.3, *б*), выполнение следующей (n + 1)-й команды начинается только после завершения предыдущей (n)-й команды. Это приводит к низкой производительности и простоям аппаратуры процессора.

Для улучшения этих характеристик используется параллельное выполнение нескольких команд путем совмещения в каждом такте различных этапов их обработки (рис. 1.3, *в*). После выборки *n-й* команды

во 2-м такте идет ее декодирование и выборка (n + 1)-й команды. В третьем такте выполняется n-я команда, декодируется (n + 2)-я и осуществляется выборка (n + 3)-й команды и т.д. Такая организация работы процессора называется **конвейерной обработкой (конвейером команд).**

Совмещенные принципы обработки (конвейер команд) существенно увеличивают пропускную способность процессора.

Приостанов работы конвейера вызывает любая команда условного перехода в программе или взаимозависимость команд, т.е. использование следующей командой результатов предыдущей команды.

Конечно, рассмотренный нами процессор является гипотетическим. В реальных процессорах конвейер обработки команд сложнее и включает большее количество ступеней. Причина увеличения длины конвейера заключается в том, что многие команды являются довольно сложными и не могут быть выполнены за один такт процессора, особенно при высоких тактовых частотах. Поэтому каждая из четырех стадий обработки команд (выборка, декодирование, выполнение и запись) может состоять из нескольких ступеней конвейера. Собственно, длина конвейера – это одна из наиболее значимых характеристик любого процессора. Чем больше длина конвейера, тем большую частоту можно использовать в процессоре.

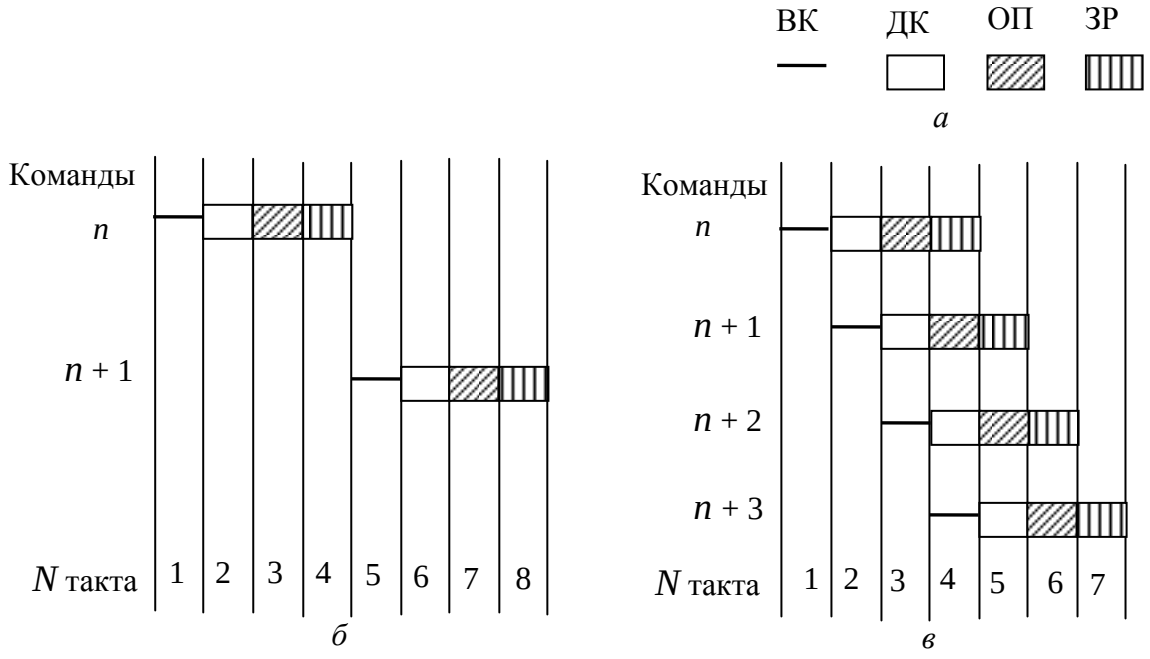


Рис. 1.3. Временные диаграммы обработки команд в процессоре:
а – этапы выполнения команды; *б* – последовательное выполнение команд;
в – совмещенное выполнение команд (конвейеризация)

Для обеспечения непрерывности вычислительного процесса в структуре ЦП используется блок прогнозирования переходов и устройство выполнения переходов.

1.1.2. Суперскалярная обработка

Смысл термина **«суперскалярная обработка»** заключается в том, что в аппаратуру процессора закладываются средства, позволяющие одновременно выполнять две или более скалярные операции, т.е. команды обработки пары чисел. Суперскалярная архитектура базируется на **многофункциональном параллелизме** и позволяет увеличить производительность компьютера пропорционально числу одновременно выполняемых операций.

Реализация суперскалярной обработки заключается в чисто аппаратном механизме выборки из буфера инструкций (или кэша инструкций) несвязанных команд и параллельном запуске их на исполнение.

Суперскалярная аппаратура динамически строит план вычислений на основе последовательного кода программы. Хотя такой подход и увеличивает сложность физической реализации, скалярный процессор создает план, используя преимущества тех факторов, которые могут быть определены только во время выполнения.

Этот метод хорош тем, что он «прозрачен» для программиста, составление программ для подобных процессоров не требует никаких специальных усилий, ответственность за параллельное выполнение операций возлагается в основном на аппаратные средства.

Суперскалярная обработка широко используется в современных процессорах корпораций Intel, Advanced Micro Devices (AMD), International Business Machines (IBM), Sun Microsystems/Oracle и др.

1.1.3. Архитектура SISD

Архитектура SISD породила целый ряд архитектур: CISC, RISC, VLIW и EPIC-концепцию (рис. 1.4).

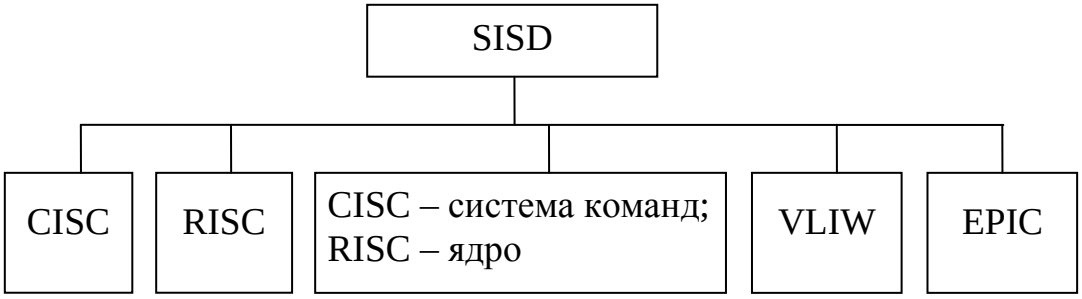


Рис. 1.4. Классификация архитектуры SISD

CISC-архитектура

Компьютеры с CISC (Complex Instruction Set Computer)-архитектурой имеют комплексную (полную) систему команд, под управлением которой выполняются всевозможные операции типа «память – память», «память – регистр», «регистр – память», «регистр – регистр».

CISC-архитектура появилась еще на заре вычислительной техники. Лидером в разработке микропроцессоров с полным набором команд считается компания Intel со своей серией процессоров x86, Pentium, Intel Core и др. Эта архитектура, получившая название x86, является практически стандартом на рынке микропроцессоров.

Данная архитектура характеризуется:

- большим числом команд (более 200);
- переменной длиной команд (от 1 до 13 байт);
- значительным числом способов адресации и форматов команд;
- наличием сложных команд и многотактностью их выполнения;
- наличием микропрограммного управления для сложных команд.

На мировых рынках полная система команд x86 представлена в процессорах фирм Intel, AMD, VIA Technologies и др.

RISC-архитектура

Компьютеры с RISC (Reduced Instruction Set Computer)-архитектурой содержат набор простых, часто употребляемых в программах команд. Основными являются операции типа «регистр – регистр».

Понятие RISC в современном его понимании оформилось на базе трех исследовательских проектов компьютеров: процессора 801 компании IBM, процессора RISC университета Беркли и процессора MIPS Стенфордского университета. Простота архитектуры и ее эффектив-

ность, подтвержденная этими проектами, вызвали большой интерес в компьютерной индустрии, и с 1986 г. началась активная промышленная реализация архитектуры RISC. Отличительные черты данной архитектуры:

- сокращенное число команд;
- большинство команд выполняется за один машинный такт;
- постоянная длина команд;
- небольшое количество способов адресации и форматов команд;
- для простых команд нет необходимости в использовании микропрограммного управления;
- большое число регистров внутренней памяти процессора.

Исходя из перечисленных характеристик, компьютеры с RISC-архитектурой «обязаны» иметь преимущество в производительности по сравнению с CISC-компьютерами.

В настоящее время основными разрабочиками RISC-процессоров являются корпорации Sun/Oracle (Ultra Sparc T1, T2), IBM (POWER 6, 6+, 7, 8, Cell). Эти процессоры используются в высокопроизводительных компьютерах (рабочих станциях, серверах, суперкомпьютерах).

Для мобильных устройств (карманных ПК, смартфонов, коммуникаторов) наибольшее распространение получили RISC-процессоры семейства ARM (корпорация ARM Ltd, Великобритания).

Уступая во многом RISC, процессоры с системой команд x86 сохранили лидерство на рынке персональных систем за счет постоянной модернизации системы команд, нацеленной на увеличение производительности процессоров, а также за счет того, что программное обеспечение, разработанное для x86-компьютеров, начиная с 1980 г., способно функционировать и на современных компьютерах с этой архитектурой. В свою очередь, достоинства RISC-процессоров укрепили их позиции на более молодом рынке высокопроизводительных машин (рабочих станций, серверов).

В начале 90-х гг. между представителями этих архитектур началась острая конкуренция за превентивное улучшение характеристик, в первую очередь производительности и ее отношения к трудоемкости разработки процессоров. Создатели CISC- и RISC-процессоров нередко боролись с конкурентами, заимствуя их удачные решения. Например, компания Intel реализовала в процессоре Pentium Pro (шестое поколение P6 процессоров Intel) RISC-подобную организацию вычислений. В P6 изощренно построенный декодер транслирует сложные команды x86 в более короткие и простые RISC-микрокоманды. В архитектуре P6 RISC-решения впервые в семействе x86 перестали быть лишь дополне-

нием исконных CISC-средств повышения производительности. Поэтому частица Pro в названии первого процессора этой серии обозначает «Полноценная RISC-архитектура» (Precision RISC Organization). На рис. 1.4 подобная архитектура вынесена в отдельный класс архитектур.

VLIW-архитектура

VLIW-архитектура связана с кардинальной перестройкой всего процесса трансляции и исполнения программ. Уже на этапе подготовки программы компилятор группирует несвязанные операции в пакеты, содержимое которых строго соответствует структуре процессора. Сформированные пакеты операций преобразуются компилятором в командные слова, которые по сравнению с обычными инструкциями выглядят очень большими. Отсюда и название этих суперкоманд, и соответствующей им архитектуры – VLIW (Very Long Instruction Word – очень длинное командное слово). По идее, затраты на формирование суперкоманд должны окупаться скоростью их выполнения и простотой аппаратуры процессора, с которого снята вся «интеллектуальная» работа по поиску параллелизма несвязанных операций.

Компилятор VLIW, в отличие от суперскалярной обработки, производит статический анализ программы и создает точный план того, как процессор будет выполнять программу: указывается, когда будет выполнена каждая операция, какие функциональные устройства будут работать и какие регистры будут содержать операнды.

Компилятор VLIW передает план вычисления аппаратному обеспечению, которое, в свою очередь, выполняет указанный план. Этот план позволяет VLIW использовать относительно простое аппаратное обеспечение, способное добиться высокого уровня параллелизма на уровне команд.

Однако даже при небольшом изменении начальных данных путь выполнения программы сколь угодно сильно изменяется.

VLIW-архитектура в свое время использовалась в RISC-процессорах семейств PA-8000, 9000 корпорации HP (Hewlett Packard), сейчас применяется в отечественных процессорах семейства «Эльбрус», а также в процессорах цифровой обработки сигналов различных фирм производителей.

Аббревиатуры рассмотренных архитектур CISC, RISC, VLIW в настоящее время обозначают только идеализированные концепции. Реальные микропроцессоры трудно классифицировать. Современные микропроцессоры, причисляемые к RISC, сильно отличаются от первых процессоров RISC-архитектуры. То же относится и к CISC. Просто

в наиболее совершенных процессорах заложено множество удачных идей, вне зависимости от их принадлежности к какой-либо архитектуре.

Концепция EPIC

Тенденции, заложенные в P6, получили развитие в концепции EPIC. Концепция EPIC (Explicity Parallel Instruction Computing – вычисления с явным параллелизмом команд, где «явным» означает явно указанным при трансляции) разработана совместно фирмами Intel и Hewlett Packard и имеет ту же значимость, что и CISC- и RISC-архитектуры.

Концепция реализации параллелизма на уровне команд (EPIC) определяет новый тип архитектуры, способной конкурировать по масштабам влияния с RISC. Эта идеология направлена на то, чтобы упростить аппаратное обеспечение и, в то же время, извлечь как можно больше «скрытого параллелизма» на уровне команд, чем это можно сделать при реализации VLIW и суперскалярных стратегий, используя большую ширину «выдачи» команд и длинные (глубокие) конвейеры.

Одна из целей, которые ставили перед собой разработчики при создании EPIC, состояла в том, чтобы сохранить реализованный во VLIW принцип статического создания плана вычислений, но, в то же время, обогатить его возможностями, аналогичными возможностям суперскалярного процессора, позволяющими новой архитектуре лучше учитывать динамические факторы, традиционно ограничивающие параллелизм, свойственный VLIW. EPIC предоставляет динамические механизмы на уровне аппаратуры так, что компилятор может управлять такими средствами, применяя их выборочно, где это возможно. Столь широкие возможности помогают компилятору использовать правила управления этими механизмами более оптимально, чем это позволяет аппаратура.

Концепция EPIC, согласно Intel и HP, обладает достоинствами VLIW, но не обладает ее недостатками.

EPIC-технология с явным заимствованием лучших идей из CISC- и RISC-архитектур использована в 64-разрядной интеловской архитектуре (IA-64) процессоров Itanium, Itanium2.

Особенности IA-64:

- использование новой системы команд, разработанный Intel и HP;
- большое количество регистров (128 64-разрядных регистров общего назначения);
- использование простых инструкций, сгруппированных по три, одинаковой длины, образующих длинные командные слова LIW (long instruction words);

- переупорядочиванием и оптимизацией команд, так же как и во VLIW, занимается компилятор, а не процессор;
- команды из разных ветвей узлового ветвления снабжаются предикатными полями (полями условий) и запускаются параллельно;
- выборка данных по предположению (выборка данных до того, как они потребуются, т.е. заранее);
- масштабируемость архитектуры до большого количества функциональных устройств.

Процессор Itanium не только реализует новые возможности 64-разрядной архитектуры, но и обладает аппаратной совместимостью с набором команд IA-32.

1.1.4. SIMD-архитектура

Как было сказано выше, архитектура SIMD связана с параллельной обработкой потоков данных. Существуют несколько способов (рис. 1.5) реализации этой архитектуры: матричная структура процессора, векторно-конвейерная, технология MMX и потоковые SIMD-расширения в интеловских процессорах.

Суть **матричной структуры** заключается в том, что имеется множество процессорных элементов, исполняющих одну и ту же команду над различными элементами матрицы, объединенных коммутатором. Основная проблема заключается в программировании обмена данными между процессорными элементами через коммутатор.

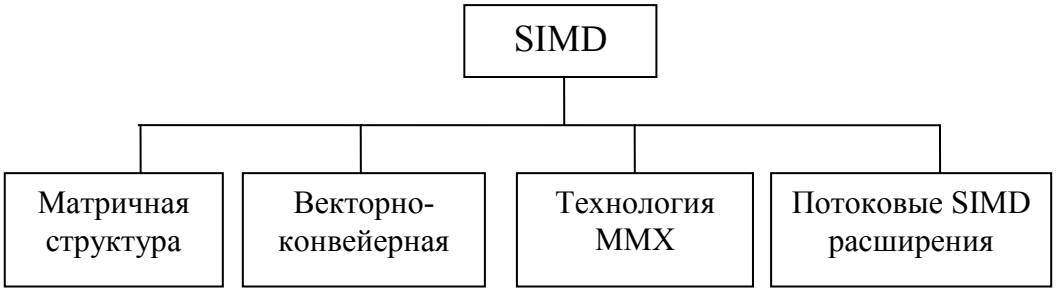


Рис. 1.5. Классификация способов организации SIMD-архитектуры

В отличие от матричной **векторно-конвейерная структура** процессора содержит конвейер операций, на котором обрабатываются параллельно элементы векторов и полученные результаты последовательно записываются в единую память. При этом отпадает необходимость в коммутаторе процессорных элементов, служащем камнем преткновения в матричных компьютерах.

Общим для всех векторных компьютеров является наличие в системе команд векторных операций, допускающих работу с векторами определенной длины. В таких компьютерах операции с векторами обычно выполняются над содержимым векторных регистров.

Еще одним примером реализации SIMD-архитектуры является **технология MMX**, которая существенно улучшила архитектуру микропроцессоров фирмы Intel (Pentium MMX). Она разработана для ускорения выполнения мультимедийных и коммуникационных программ. Команды MMX выполняют одну и ту же функцию с различными частями данных, например, 8 байт графических данных передаются в процессор как одно упакованное 64-разрядное число и обрабатываются одной командой.

Следующим шагом по пути использования SIMD-архитектуры в микропроцессорах фирмы Intel (Pentium III) явились **потоковые SIMD-расширения – Streaming SIMD Extension (SSE),** которые реализуют новые SIMD-инструкции, оперирующие со специальными 128-битными регистрами. Каждый из этих регистров может хранить несколько упакованных целочисленных или вещественных данных. Таким образом, выполняя операцию над содержимым двух регистров под управлением команды SSE, процессор может обработать несколько пар операндов одновременно.

Несколько раньше то же самое было сделано фирмой AMD – расширение 3DNow!, которое было реализовано уже в процессорах K6-2 с введением новых инструкций, оперирующих с 64-битными регистрами.

Данное направление получило развитие и в следующих поколениях процессоров корпораций Intel и AMD. Современные процессоры Intel поддерживают потоковые расширения SSE, SSE2, SSE3, SSSE3, SSE4, AVX, AVX2, MIC и т.д.

1.1.5. Многоядерные структуры процессора и многопотоковая обработка команд

Корпорация Intel, лидер в разработке микропроцессоров с x86 архитектурой, ежегодно на протяжении долгого времени увеличивала производительность своих процессоров преимущественно за счет увеличения тактовой частоты и использования гиперконвейерной технологии выполнения команд, что, в свою очередь, значительно увеличивало энергопотребление и соответственно количество выделяемой процессором тепловой энергии. Это привело к тому, что компания уперлась в энергетический предел, ограничивающий возможности наращивания производительности процессорных кристаллов традиционными спосо-

бами. Перед компанией Intel остро встала проблема разрешения противоречия между производительностью процессора и энергопотреблением.

Использование многоядерных структур процессора является одним из путей решения этой проблемы. Совмещение в одном процессоре двух вычислительных ядер позволяет удерживать рассеиваемую им мощность в допустимых пределах за счет сравнительно незначительного понижения тактовой частоты ядер: при снижении рабочей частоты на 20 % производительность ядра падает примерно на 13 %, а энергопотребление – на 50 %. При этом двухъядерный процессор все равно существенно выигрывает в производительности (при тех же условиях до 70 %) за счет увеличения количества команд, выполняемых в процессоре за один такт, но для этого необходимо на программном уровне обеспечить загрузку обоих ядер, для чего требуется соответствующая оптимизация программного кода.

Первыми стали использовать двухъядерные структуры разработчики RISC-процессоров:

- компания IBM (процессоры Power 4, 5, Power PC G5);
- Sun Microsystems (процессор Ultra Sparc IV).

В настоящее время выпускается достаточно большое количество типов многоядерных процессоров различных фирм производителей с 2, 4, 6, 8, 12 ядрами. Можно сказать, что в развитии вычислительной техники с 2005 г. наступила эра использования многоядерных структур процессоров.

Другим направлением развития микропроцессорной индустрии на ближайшие годы будет многопоточность. Двупотоковая обработка команд на одном процессоре (ядре) основывается на том, что в каждый момент времени только часть ресурсов процессора (ядра) используется при выполнении программного кода. Неиспользуемые ресурсы также можно загрузить работой, например задействовать для параллельного выполнения еще одного приложения. В этом случае операционная система (ОС) и приложения «видят» именно два логических процессора (ядра) и могут распределять работу между ними, как и в случае полноценной двухпроцессорной системы (рис. 1.6).

Для того чтобы использовать технологии многопоточности, необходимы эффективные компиляторы, которые разработаны и поставляются вместе с микропроцессорами.

Технологии многопоточности в настоящее время используются различными фирмами:

- Intel – технология Hyper-Threading (HT), технология Simultaneous multithreading (SMT);
- Sun/Oracle– технология Chip Multithreading (CMT);

- Fujitsu Siemens Computer – технология Vertical Multithreading (VMT).

Применение многоядерной структуры одновременно с технологией многопоточности увеличивает количество используемых логических процессоров (ядер) в 2 раза (Core i7, Itanium 2, Xeon), в 4 раза (Ultra SPARC T1), в 8 раз (Ultra SPARC T2), что существенно увеличивает производительность физического процессора.

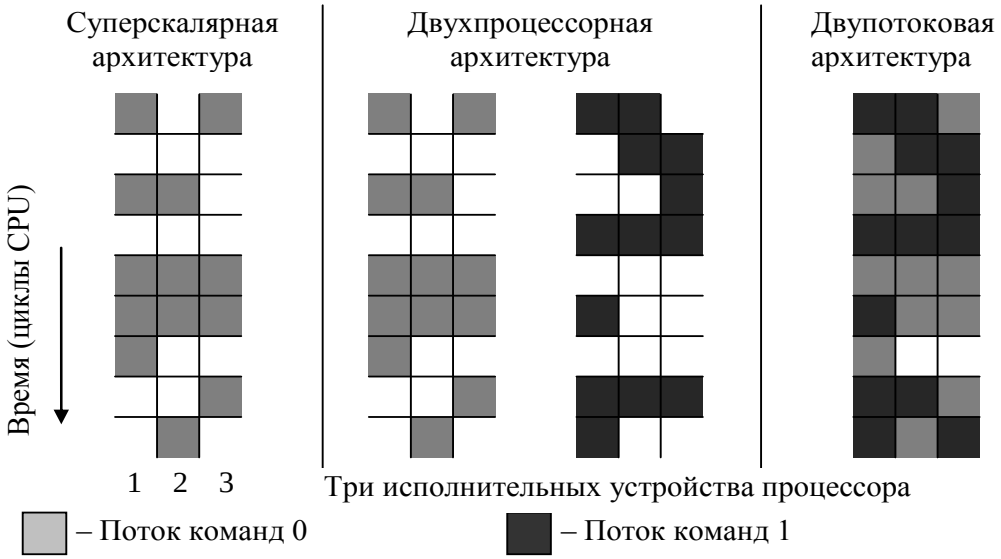


Рис. 1.6. Многопоточность
в сравнении с другими способами обработки команд

1.2. Технические и эксплуатационные характеристики ЭВМ

Производительность компьютера

Основным техническим параметром ЭВМ является её **производительность**. Этот показатель определяется архитектурой процессора, иерархией внутренней и внешней памяти, пропускной способностью системного интерфейса, системой прерывания, набором периферийных устройств в конкретной конфигурации, совершенством ОС и т.д.

Различают следующие виды производительности:

- **пиковая** (предельная) – это производительность процессора без учета времени обращения к оперативной памяти (ОП) за операндами;
- **номинальная** – производительность процессора с ОП;
- **системная** – производительность базовых технических и программных средств, входящих в комплект поставки ЭВМ;
- **эксплуатационная** – производительность на реальной рабочей нагрузке, формируемой в основном используемыми пакетами прикладных программ общего назначения.

Методы определения производительности разделяются на три основных группы:

- **расчетные**, основанные на информации, получаемой теоретическим или эмпирическим путем;
- **экспериментальные**, основанные на информации, получаемой с использованием аппаратно-программных измерительных средств;
- **имитационные,** основанные на моделировании и применяемые для сложных ЭВМ.

Основные единицы оценки производительности:

- **абсолютная**, определяемая количеством элементарных работ, выполняемых в единицу времени;
- **относительная**, определяемая для оцениваемой ЭВМ относительно базовой в виде индекса производительности.

Для каждого вида производительности применяются следующие традиционные методы их определения.

Пиковая производительность (быстродействие) определяется средним числом команд типа «регистр–регистр», выполняемых в одну секунду, без учета их статистического веса в выбранном классе задач.

Номинальная производительность (быстродействие) определяется средним числом команд, выполняемых подсистемой «процессор–память» с учетом их статистического веса в выбранном классе задач. Она рассчитывается, как правило, по формулам и специальным методикам, предложенным для процессоров определенных архитектур, и измеряется с помощью разработанных для них измерительных программ, реализующих соответствующую эталонную нагрузку.

Для данных типов производительностей используются следующие единицы измерения:

- MIPS (Mega Instruction Per Second) – миллион команд в секунду;
- MFLOPS (Mega Floating Operations Per Second) – миллион операций над числами с плавающей запятой в секунду;
- GFLOPS (Giga Floating Operations Per Second) – миллиард операций над числами с плавающей запятой в секунду;
- TFLOPS (Tera Floating Operations Per Second) – триллион операций над числами с плавающей запятой в секунду;
- PFLOPS (Peta Floating Operations Per Second) – квадриллион операций над числами с плавающей запятой в секунду.

Системная производительность измеряется с помощью синтезированных типовых (тестовых) оценочных программ, реализованных на унифицированных языках высокого уровня. Унифицированные тестовые программы используют типичные алгоритмические действия, ха-

рактерные для реальных применений, и штатные компиляторы ЭВМ. Они рассчитаны на использование базовых технических средств и позволяют измерять производительность для расширенных конфигураций технических средств. Результаты оценки системной производительности ЭВМ конкретной архитектуры приводятся относительно базового образца, в качестве которого используются ЭВМ, являющиеся промышленными стандартами систем ЭВМ различной архитектуры. Результаты оформляются в виде сравнительных таблиц, двумерных графиков и трехмерных изображений.

Эксплуатационная производительность оценивается на основании использования данных о реальной рабочей нагрузке и функционировании ЭВМ при выполнении типовых производственных нагрузок в основных областях применения. Расчеты делаются главным образом на уровне типовых пакетов прикладных программ текстообработки, систем управления базами данных, пакетов автоматизации проектирования, графических пакетов и т.д.

Была создана целая **процедура тестирования True Performance Initiative** (процедура измерения реальной производительности – TPI). Методика TPI состоит в измерении эксплуатационной производительности в трех разделах: Productivity – программные приложения; Visual Computing – компьютерная визуализация; Gaming – компьютерные игры.

Энергоэффективность процессора

В последнее время при сравнении процессоров пользуются отношением производительности к энергопотреблению, которое получило название **энергоэффективность процессора**. Разработчики процессоров предложили оценивать производительность (P) как произведение тактовой (рабочей) частоты процессора (f) на величину k, определяющую количество инструкций, исполняемых процессором за один такт:

$$P = f \cdot k.$$

Получается, что увеличить производительность можно, поднимая частоту и/или увеличивая количество инструкций, выполняемых за один такт. Первый подход ведет к увеличению энергопотребления, а второй требует использования определенной микроархитектуры процессора (многоядерной), в которой заложены различные технологии, направленные на повышение количества инструкций, выполняемых процессором за один такт.

Что касается энергопотребления (W), то оно вычисляется как произведение тактовой частоты (f) процессора на квадрат напряжения U, при котором функционирует процессорное ядро, и некоторую величину Cd (динамическая емкость), определяемую микроархитектурой процессора и зависящую от количества транзисторов в кристалле и их активности во время работы процессора:

$$W = f \cdot U^2 \cdot Cd.$$

Из приведенных формул вытекает следующее соотношение, определяющее энергоэффективность процессора:

$$P/W = k / (U^2 \cdot Cd).$$

Из формулы следует, что для получения наилучшего показателя разработчикам необходимо работать над оптимизацией микроархитектуры с целью улучшения функциональности исполнительных блоков, при этом не допуская чрезмерного увеличения динамической емкости. Что касается тактовой частоты, то, как показывают приведенные выкладки, на рассматриваемое соотношение она вообще не влияет. Напряжение питания ядра зависит не столько от микроархитектуры, сколько от технологических особенностей изготовления процессора.

Любой современный кристалл процессора состоит из огромного количества транзисторов, исчисляемого миллионами, необходимого для достижения высокой производительности процессора. Уменьшение размеров транзистора ведет к уменьшению напряжения питания, что, в свою очередь, снижает энергопотребление, к увеличению скорости работы и плотности размещения транзисторов на кристалле. Поэтому со времени создания первой интегральной микросхемы в 1959 г. развитие микроэлектроники идет по направлению уменьшения размеров транзисторов и одновременного увеличения плотности их размещения на кристалле. Для оценки этих параметров была введена специальная характеристика «**Норма технологического процесса производства полупроводниковых кристаллов**», измеряемая в нанометрах (нм). В недалеком прошлом (конец 90-х гг.) кристаллы процессоров изготавливались по 130 нм нормам, затем по 90 нм, 65 нм. С 2008 г. используются 45 нм, с 2010 г. – 32 нм, с 2012 г. – 22 нм, а с 2014 г. – 14 нм нормы технологического процесса. Спроектированный в Intel по 45 нм нормам транзистор примерно на 20 % опережает своего 65 нм собрата по скоростным характеристикам и оказывается примерно на 30 % экономичнее с точки зрения затрат энергии на переключение.

Часто вместо характеристики **энергопотребление** используют характеристику **рассеиваемая тепловая мощность** процессора. Для этого используется специальный термин **TDP**, который расшифровывается как **термопакет** (thermal design package) – это величина, показывающая, на отвод какой тепловой мощности должна быть рассчитана система охлаждения процессора.

Как правило, TDP показывает не максимальное теоретическое тепловыделение процессора, а типичное тепловыделение в реальных приложениях. Иногда, при длительных нагрузках на процессор (например, при кодировании видео), температура процессора может превысить заданный TDP. В этих случаях современные процессоры или дают сигнал выключения компьютера, или переходят в так называемый режим тротлинга (trottling), когда процессор пропускает часть циклов.

К другим технико-эксплуатационным характеристикам ЭВМ относятся:

- разрядность обрабатываемых слов и кодовых шин интерфейса;
- типы системного и локального интерфейсов;
- тип и емкость оперативной памяти;
- тип и емкость накопителя на жестком магнитном диске;
- тип и емкость кэш-памяти;
- тип видеоадаптера и видеомонитора;
- наличие средств для работы в компьютерной сети;
- наличие и тип программного обеспечения;
- надежность ЭВМ;
- стоимость;
- габариты и масса.

1.3. Классификация ЭВМ

1.3.1. Классификация ЭВМ по назначению

По назначению ЭВМ можно разделить на три группы: **универсальные** (общего назначения), **проблемно-ориентированные** и **специализированные.**

Универсальные ЭВМ предназначены для решения самых различных видов задач: научных, инженерно-технических, экономических, информационных, управленческих и др. В качестве универсальных ЭВМ используются различные типы компьютеров, начиная от суперЭВМ и кончая персональными ЭВМ. Причем одни универсальные ЭВМ могут работать в многопользовательском режиме (в вычислительных центрах коллективного пользования, в локальных компьютерных сетях и т.д.), другие – в однопользовательском режиме.

Проблемно-ориентированные ЭВМ служат для решения более узкого круга задач, связанных, как правило, с управлением технологическими объектами, автоматизированным проектированием, разведкой и добычей нефти, банковским делом, издательской деятельностью и т.д.

Специализированные ЭВМ используются для решения еще более узкого круга задач или реализации строго определенной группы функций. Такая узкая ориентация ЭВМ позволяет четко специализировать их структуру, во многих случаях существенно снизить их сложность и стоимость при сохранении высокой производительности и надежности их работы.

1.3.2. Классификация ЭВМ по функциональным возможностям

По функциональным возможностям и размерам ЭВМ можно разделить на **суперЭВМ**, **большие** и **микроЭВМ**.

Функциональные возможности ЭВМ обусловливаются основными технико-эксплуатационными характеристиками.

Исторически первыми появились большие ЭВМ, элементная база которых прошла путь от электронных ламп до интегральных схем со сверхвысокой степенью интеграции.

Большие ЭВМ

Большие ЭВМ за рубежом часто называют **мэйнфреймами** (Mainframe). Мэйнфрейм – это высокопроизводительная вычислительная система с большим объемом оперативной и внешней памяти, поддерживающая многопользовательский и многозадачный режимы работы.

Особенности и характеристики современных мэйнфреймов

К ним относятся:

1. **Высокая надежность** (среднее время наработки на отказ оценивается в 12–15 лет) – результат почти 50-летнего совершенствования мэйнфреймов.

2. **Повышенная устойчивость** систем. Мэйнфреймы могут обнаруживать, исправлять и изолировать большинство аппаратных и программных ошибок.

3. **Целостность данных.** В мэйнфреймах используется память с исправлением ошибок.

4. **Рабочая нагрузка мэйнфреймов** может составлять 80–95 % от их пиковой производительности.

5. **Высокая пропускная способность** подсистемы ввода/вывода (канальная архитектура).

6. **Масштабирование** может быть как вертикальным, так и горизонтальным. Вертикальное масштабирование обеспечивается наращиванием до 64 центральных процессоров в одном компьютере. Горизонтальное – реализуется объединением компьютеров в многомашинный (до 32 машин) кластер, выглядящий, с точки зрения пользователя, единым компьютером.

7. **Доступ к данным.** При централизованной обработке информации данные хранятся на одном компьютере, прикладные программы не нуждаются в сборе исходной информации из множества источников (как при распределенной обработке), не требуется дополнительное дисковое пространство для их временного хранения, не возникают сомнения в их актуальности. Все это ведет к снижению стоимости и повышению эффективности обработки.

8. **Защита.** Встроенные аппаратные и программные средства защиты, такие как криптографические устройства, программные продукты защиты операционных систем, обеспечивают совершенную защиту информации.

9. **Непрекращающаяся совместимость –** до сих пор в мэйнфреймах используются приложения, написанные в 70-е гг. Историю полупроводниковых мэйнфреймов принято отсчитывать с появления в 1964 г. универсальной компьютерной системы IBM System/360. За последние десятилетия мэйнфреймам неоднократно предрекали скорую кончину, однако время доказало, что сбить с ног этих «старожилов» не так-то просто.

Централизованная архитектура остается востребованной, несмотря на преобладание в современном крупном бизнесе распределенных вычислительных систем. Сторонники мэйнфреймов утверждают, что такая архитектура обеспечивает нормальное функционирование системы при 100%-й нагрузке процессоров, тогда как производительность стандартных серверов ощутимо снижается уже при 65%-й нагрузке.

За долгие годы существования мэйнфреймов для них было разработано великое множество прикладного программного обеспечения, однако лучшим доказательством заинтересованности рынка является разработка и выпуск новых моделей этого класса.

До сегодняшнего дня бесспорным лидером в производстве мэйнфреймов является корпорация IBM, начиная от серии System/360, затем 370, 390 и до серии z Series. Первые мэйнфреймы этой серии были z800, 890, 900, 990. В 2005 г. IBM объявила о выпуске новых машин z Series

семейства «Z». Очень удачным экземпляром этого семейства была машина z9. В 2008 г. компания IBM выпустила в свет мэйнфрейм System z10 Enterprise Class, представляющий собой 64-процессорную систему, в которой установлены новые процессоры с четырьмя ядрами и частотой 4,4 ГГц. Мэйнфрейм System z10 поддерживает операционные системы z/OS, z/OSe, z/VM, z/VSE, Linux и может обслуживать от сотен до миллионов пользователей в зависимости от приложений.

В 2015 г. компания IBM презентовала новый мэйнфрейм z13, на создание которого ушло 5 лет и около миллиарда долларов США. В процессе работы были использованы инновационные технологии, более 500 новых патентов. Максимальная конфигурация (рис. 1.7) системы z13 включает 141 процессор и 10 ТБ ОЗУ. Процессор содержит 8 ядер, тактовая частота работы – 5 ГГц, технология изготовления – 22 нм.

Рис. 1.7. Общий вид мэнфрейма IBM z13

Основными направлениями эффективного применения мэйнфреймов являются пакетная обработка заданий (когда компьютер выполняет работу без участия человека) и обработка заданий в реальном времени (On-line), например транзакционные системы, такие как система приобретения железнодорожных билетов, система оплаты по кредитной карте и т.п.

Система z13 может обрабатывать около 2,5 миллиардов транзакций в день, обладает встроенной функцией шифрования и анализа проводимых транзакций, что позволяет выявлять случаи мошенничества в режиме реального времени для 100% бизнес-операций, моментально предоставляя пользователю аналитические данные для оценки операции.

При использовании мэйнфрейма можно запустить вплоть до 8000 виртуальных серверов. Это около 50 серверов из расчета на один процессор. Корпорация IBM выпустила новый релиз ОС – z/OS, в которой используется новый тип аналитики, есть поддержка Linux и систем хранения данных.

В последние годы наметился повышенный интерес крупного бизнеса к мэйнфреймам как центрам IT-инфраструктуры. Практика подтверждает: почти все мировые банки из списка Fortune Top 25 используют System z для обработки данных.

СуперЭВМ

СуперЭВМ – мощные, высокоскоростные вычислительные машины (системы) с производительностью от десятков триллионов (GFLOPS) до нескольких десятков квадриллионов (PFLOPS) операций с плавающей запятой в секунду. СуперЭВМ выгодно отличаются от больших универсальных ЭВМ по быстродействию числовой обработки, а от специализированных машин, обладающих высоким быстродействием в сугубо ограниченных областях, – возможностью решения широкого класса задач с числовыми расчетами.

В настоящее время развитие суперкомпьютеров идет по следующим направлениям: векторно-конвейерные компьютеры, параллельные компьютеры с общей памятью, массивно-параллельные системы с распределенной памятью, кластерные системы.

В 2009 г. был преодолен порог производительности суперкомпьютеров в 1 PFLOPS (10^{15} FLOPS). На сегодняшний день в мире насчитывается уже достаточно большое количество суперкомпьютеров, начиная от простых (офисных и персональных) и кончая мощными массивно-параллельными и кластерными системами.

Два раза в год формируется официальный список пятисот самых мощных суперкомпьютеров мира – Top500. В ноябре 2015 г. список Top500 (табл. 1.1.) возглавила третий год подряд система Tianhe-2 (рис. 1.8) китайского национального университета оборонных технологий с неизменной за всё это время производительностью (33,9 Пфлопс).

Рис. 1.8. Общий вид системы Tianhe-2

Обновились лишь шестая и восьмая строчки, отныне занятые системой Trinity для лабораторий Los Alamos и Sandia (8,1 Пфлопс), и системой Hazel-Hen для исследовательского центра HLRS в Штутгарте (5,6 Пфлопс), обе на базе архитектуры Cray XC. Шесть систем списка TOP10 инсталлированы ещё в 2011-12 гг., лидер Tianhe-2 в 2013 г., и только три в 2015 г. Сложившаяся ситуация чётко отражает начавшийся в 2008 г. тренд на снижение активности в десятке лидеров последних десятилетий.

Гораздо сильнее изменения на региональном уровне. Лидером обновлений является Китай, практически утроивший число систем в последнем списке TOP500, и, что не менее важно, представленный большим числом местных разработчиков высокопроизводительных систем. Число суперкомпьютеров, инсталлированных в США, напротив, снизилось до 200 (в предыдущем июльском 45-м рейтинге их было 231), продемонстрировав тем самым минимальное присутствие в списке с момента создания TOP500 в 1993 г.

Число европейских систем в списке снизилось за полгода со 141 до 108, зато системы азиатского региона сделали за то же время гигантский скачок со 107 до 173 систем, при этом 109 из них работают в Китае (полгода назад в списке было только 37 китайских систем).

Таблица 1.1

Рейтинг 10 самых производительных суперЭВМ

RANK	SITE	SYSTEM	CORES	RMAX (PFLOP/S)	RPEAK (PFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3 120 000	33.9	54.9	17 808
2	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560 640	17.6	27.1	8 209
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1 572 864	17.2	20.1	7 890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705 024	10.5	11.3	12 660
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786 432	8.6	10.1	3 945
6	DOE/NNSA/LANL/SNL United States	**Trinity** - Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Aries interconnect Cray Inc.	301 056	8.1	11.1	
7	Swiss National Supercomputing Centre (CSCS) Switzerland	**Piz Daint** - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc.	115 984	6.3	7.8	2 325
8	HLRS - Höchstleistungsrechenzentrum Stuttgart Germany	**Hazel Hen** - Cray XC40, Xeon E5-2680v3 12C 2.5GHz, Aries interconnect Cray Inc.	185 088	5.6	7.4	
9	King Abdullah University of Science and Technology Saudi Arabia	**Shaheen II** - Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Aries interconnect Cray Inc.	196 608	5.5	7.2	2 834
10	Texas Advanced Computing Center/Univ. of Texas United States	**Stampede** - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P Dell	462 462	5.2	8.5	4 510

В некоторой степени такой расклад объясняется усилением позиций Lenovo, поглотившей недавно серверный x86-бизнес IBM, но лишь отчасти, поскольку в TOP500 сейчас насчитывается 25 систем производства Lenovo (всего три в предыдущем списке), в то время как другой китайский производитель, компания Sugon, является производителем 49 систем последнего листинга, что превышает как собственно рейтинг

IBM, снизившийся за полгода с 23 до 14,9%, так и число двухбрендовых позиций: IBM/Lenovo (девять систем) и Lenovo/IBM (пять систем). Первое место среди вендоров с долей 31,2% по-прежнему занимает HP (155 систем против 178 в прошлом рейтинге), второе место с долей 13,8% занимает Cray с неизменными 69 системами.

В последнем рейтинге TOP500 насчитывается уже 80 систем с производительностью более 1 Пфлопс, хотя в предыдущем рейтинге их было только 67. Суммарная производительность входящих в последний рейтинг систем составила 420 Пфлопс (361 Пфлопс в 45-м рейтинге и 309 Пфлопс годом ранее). Система с производительностью 204,3 Тфлопс, замыкающая нынешний список TOP500, числилась на 369-й позиции предыдущего листинга, в котором производительность последней системы составляла всего 164 Тфлопс. Тенденция последних шести лет, когда ежегодный суммарный прирост производительности в рейтинге TOP500 составляет не более 55% (для сравнения: до 90% в период 1994-2008 гг.), в основном является результатом медленного обновления верхней части списка, что, в свою очередь, можно расценивать как результат разнонаправленных маркетинговых тенденций в сегменте крупных суперкомпьютеров и сегменте средних и небольших систем высокопроизводительных вычислений (HPC).

В последний рейтинг вошли 104 системы с сопроцессорными и ускорительными технологиями, полгода назад в предыдущем TOP500 их насчитывалось всего 90. Первый (Tianhe-2) и последний (Stampede) суперкомпьютеры первой десятки рейтинга используют для ускорения вычислений сопроцессоры Intel Xeon Phi, второй (Titan) и седьмой (Piz Daint) оснащены графическими акселераторами Nvidia. Четыре системы из нового рейтинга включают как чипы Intel Xeon Phi, так и GPU-ускорители Nvidia, 66 систем выполнены с применением акселераторов Nvidia, три на акселераторах ATI Radeon, 27 на сопроцессорах Intel Xeon Phi.

Из пяти сотен систем 445 (89%) выполнены на процессорах Intel (86,2% в прошлом рейтинге), при этом чипы IBM Power используются только в 26 системах (38 систем в прошлом рейтинге), а чипы AMD Opteron в 21 системе (4,2%, в прошлом рейтинге — 4,4%). В сумме 86% систем выполнены на процессорах с числом ядер шесть и более, 88% — на восьми- и более ядерных процессорах, 47% на чипах с десятью и более ядрами. Технология InfiniBand по-прежнему остаётся наиболее популярным способом внутрисистемных коммуникаций, хотя её распространённость заметно упала: с 257 систем в прошлом рейтинге до 235 в новом. Напротив, распространённость Gigabit Ethernet выросла

со 147 систем до 182, при этом в подавляющем числе случаев — 120 — речь идёт об интерфейсах 10G.

За полгода, прошедшие с момента подведения итогов 45-го рейтинга, из списка TOP500 выбыла находившаяся ранее на 466-м месте одна из суперкомпьютерных систем Санкт-Петербургского политехнического университета — «Политехник RSC PetaStream» (РСК). Второй суперкомпьютер СПбПУ «Политехник РСК Торнадо» на основе кластерной архитектуры «РСК Торнадо» с прямым жидкостным охлаждением (РСК) сейчас занимает 131-ю строчку мирового рейтинга с результатом 658 Тфлопс. Кроме того, в рейтинге присутствуют еще два суперкомпьютера производства РСК — на 256-м месте МВС-10П, установленный в МСЦ РАН, и на 348-й позиции «РСК Торнадо ЮУрГУ», эксплуатируемый с 2009 г. в Южно-Уральском государственном университете (ЮУрГУ) в Челябинске.

На сегодняшний день в TOP500 осталось семь российских систем (полгода назад их было восемь), а самый производительный российский суперкомпьютер «Ломоносов-2» («Т-Платформы») продолжает терять позиции, переместившись с 31-й строчки на 36-ю.

Если рассматривать TOP500 по вхождению в него систем российских вендоров, здесь можно отметить появление на 50-й позиции новой системы JURECA в суперкомпьютерном центре Юлиха (Германия), созданной компанией «Т-Платформы». На сегодняшний день в TOP500 входит по три системы производства РСК и «Т-Платформы», а также суперкомпьютер «Курчатовского института» (№ 258) и еще одна система производства компании «Ниагара» (№ 346).

В каких рыночных нишах востребованы суперкомпьютеры? Прежде всего, это проектирование самолетов и ракет, создание лекарств, предсказание погоды и природных катаклизмов, повышение эффективности электростанций и надежности автомобилей (преимущественно путем моделирования их столкновений) и фундаментальные научные исследования.

МикроЭВМ

МикроЭВМ по назначению можно разделить на серверы, рабочие станции, персональные компьютеры, встраиваемые и промышленные микроЭВМ (рис. 1.7).

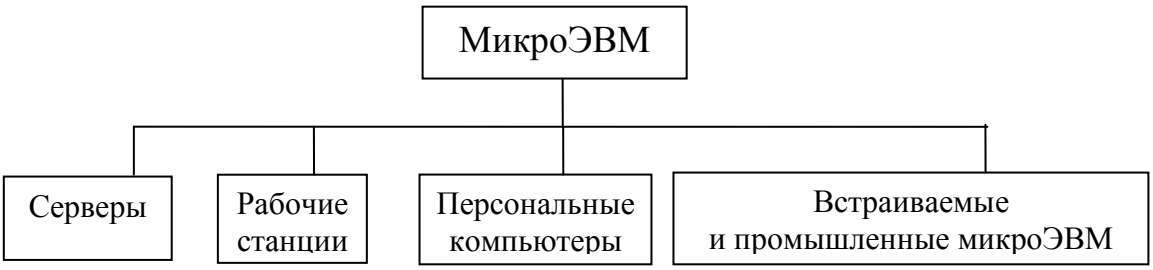


Рис. 1.7. Классификация микро-ЭВМ

Серверы

В настоящее время уже редко встретишь офис или предприятие, в котором бы не использовалась компьютерная сеть, время разрозненных персональных компьютеров давно ушло. Однако нагрузка, т.е. уровень сетевого трафика, на различные узлы в сети никогда не бывает равномерно распределенной – на пользовательских компьютерах она всегда меньше, чем на компьютерах, выполняющих служебные функции в сети, **серверах** (от англ. «serve» – служить).

Примером таких функций может быть хранение файлов и обеспечение доступа к ним пользователей (клиентов), маршрутизация потоков данных, управление печатью сетевого принтера, обработка писем электронной почты, рассылка факсов и т.д. Серверами также называются программы, выполняющие эти функции. Ниже под термином «сервер» будет пониматься в первую очередь аппаратное решение.

По функциональному назначению серверы можно подразделить (рис. 1.8) на **файл-серверы, серверы приложений** (чаще всего используются для баз данных и поддержки документооборота), **FTP-серверы** (для удаленного доступа к данным через Internet), **серверы внешних устройств** (печати, сканирования, факсимильной связи) и **Web-серверы.**

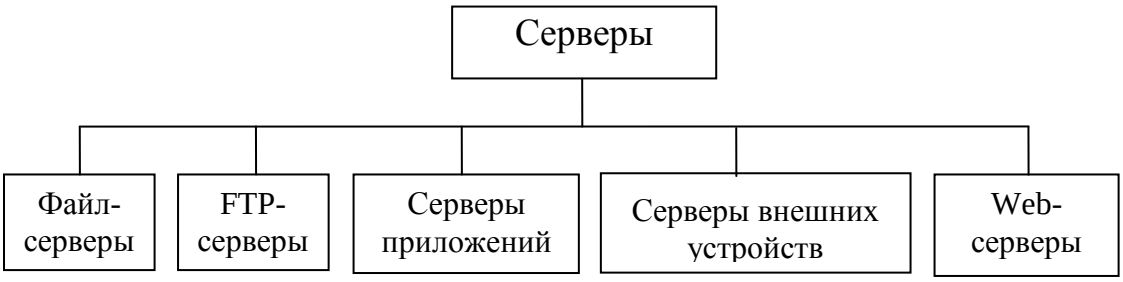


Рис. 1.8. Классификация серверов по функциональному назначению

Крупные и мелкие предприятия и офисы обладают вычислительными сетями различной мощности. Кроме того, существуют разные требования к функциям, выполняемым компьютерной сетью, если од-

ной организации достаточно иметь один файловый сервер, то для другой требуется полный спектр Internet-сервисов, таких как обеспечение получения и отправки электронной почты для всех сотрудников, хостинг (возможность размещения) Web-сайта или FTP-файлового архива. Поэтому не существует «универсального» сервера, способного выполнять любые, совершенно различные задачи одинаково быстро и эффективно.

По функциональным возможностям (мощности) серверы разделяют на **серверы начального**, **среднего** и **корпоративного уровней.** На каждом уровне используются свои способы организации серверов. Для небольшой сети (в рамках рабочих групп – 50 и менее пользователей) функции сервера могут быть возложены на мощный настольный персональный компьютер. Для среднего уровня (50–200 клиентов и малых серверов) могут быть использованы мощные рабочие станции, а для корпоративного (200 и более пользователей) – мэйнфреймы. Кроме того, для каждого уровня иерархии разрабатываются и применяются компьютеры со специальной серверной организацией.

В серверах начального уровня используются как правило от 1 до 8 процессоров, среднего уровня – от 8 до 16 процессоров, корпоративного уровня – от 32 до нескольких сотен процессоров. Количество ядер в процессорах может варьироваться от 2 до 16.

Приведенные классификации весьма условны, потому что в рамках любой серии серверов постоянно появляются модели большей мощности благодаря наращиванию ресурсов и модернизации конфигурации, причем различия внутри одной линейки компьютеров могут быть существенны.

Основными требованиями при проектировании серверов являются:

- **большая мощность** для обеспечения нормальной работы всех запускаемых приложений;
- **масштабируемость**, необходимая при увеличении компьютерной сети предприятия или круга задач, решаемых сервером;
- **отказоустойчивость** для обеспечения надежной работы всех выполняемых программ и сервисов;
- **удобный доступ** к его компонентам с возможностью оперативной или даже «горячей» (автоматической) замены, что очень важно в случае необходимости бесперебойной работы системы.

Сервер начального уровня может собрать хорошо разбирающийся в вычислительной технике человек, но наиболее надежные и сбалансированные системы выпускаются brand-name компаниями, специализирующимися на производстве серверов. Компоненты этих систем обычно хорошо подобраны друг с другом, что увеличивает эффективность их использования.

Для реализации серверов используются процессоры с архитектурами x86 (CISC), IA-64 (EPIC) и RISC. В качестве основной операционной системы (ОС) может выступать Windows-подобная или Unix-подобная ОС (Unix-серверы).

В последние годы большой популярностью пользуется **операционная система Linux** с открытым исходным кодом, которая была создана в 1991 г. в качестве версии Unix-подобной системы для ПК. Настоящий успех пришел к системе при ее использовании не на настольных ПК, а на серверах. Этому способствует мощная поддержка «китов индустрии» (IBM, HP, Dell, Sun/Oracle), вкладывающих в развитие бесплатной ОС огромные средства.

3 марта 2015 года исследовательская компания International Data Corporation (IDC) опубликовала результаты анализа глобального рынка серверов. Его объем, как выяснили эксперты, достиг рекордного значения, в чем большая заслуга Китая.

Согласно оценкам IDC, в 2014 году на мировой рынок было выпущено 9,2 млн серверов, что на 2,9% больше, чем годом ранее. Столь крупного объема не было никогда, отмечают эксперты. В деньгах серверный рынок показал 2,3-процентный рост — до $50,9 млрд.

В октябре-декабре 2014 года продажи серверов увеличились на 1,9% (до $14,5 млрд), и этот квартальный подъем оказался третьим подряд. Девятимесячную положительную динамику аналитики объясняют продолжающимся циклом обновления компьютерного оборудования в компаниях и растущими инвестициями в инфраструктуру со стороны облачных провайдеров.

Кроме того, немаловажную роль играет китайский рынок серверов, объем которого достиг рекордного уровня в $2 млрд по итогам последних трех месяцев 2014 года, увеличившись на 26,2% в сравнении с аналогичным периодом 2013-го. Четыре крупнейших вендора из Поднебесной Inspur, Huawei, Lenovo и Sugon нарастили серверную выручку более чем наполовину, говорится в отчете IDC.

Крупнейшим производителем серверов в глобальном масштабе остаётся HP, однако в 2014 году американская компания увеличила продажи систем всего на 0,8%, а рыночная доля вендора упала до 26,2% с 26,6% годом ранее.

Следом за HP идет ее соотечественница IBM, чьи позиции после продажи направления x86-решений начали стремительно слабеть. 2014 год «голубой гигант» закончил с 18,4-процентной долей, тогда как годом ранее она измерялась 25,6%. В четвертой четверти показатель при-

сутствия IBM снизился до 13,7%, и корпорация уступила второе место Dell.

Последняя в 2014 году заработала на серверах $9 млрд, подняв этот доход на 5,7% в годовом исчислении. В пятерку крупнейших производителей также вошли Cisco и Oracle, занявшие 5,7% и 4,6% рынка соответственно.

Для оценки производительности серверов используется тест TPC-C, разработанный фирмой Transaction Processing Perfomance Council (TPC), с помощью которого подсчитывается количество транзакций (взаимодействий между узлами компьютерной сети по принципу *запрос–ответ*) в минуту (tpm C) и удельная стоимость ($/tpm C) транзакций.

По результатам работы в 2012 г. около четверти серверов использовались в рамках «вычислительных облаков». Термин «облачные вычисления» (cloud computing – CC), ставший модным в последнее время, зародился еще в 1960 г., но обрел актуальность только с лавинообразным развитием Интернета, – в частности, вместе с ростом скоростей и эволюцией браузеров. **Облачные вычисления** – технология обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю как Интернет-сервис. Вся работа с данными происходит на удаленном сервере, а компьютер-клиент лишь интерактивно общается с «облачной фермой», получая от нее картинку. Рядовой пример простого CC – любая веб-почта. В сеть Интернет перебираются приложения, которые раньше представлялись в виде здоровенных программ, продававшихся на DVD: текстовые редакторы, электронные таблицы, графические пакеты, игры и т.д.

По мере того, как компании станут все шире использовать такие модели вычислений, как «облака», будет расти спрос на **микросерверы** – малогабаритные серверы с малым потреблением энергии и невысокой ценой.

Новый класс устройств – **миниатюрные домашние серверы** – становится обыденным явлением не только дома, но и в офисах малых и мобильных компаний. В настоящий момент на рынке можно найти около полудюжины производителей, предлагающих портативные и не очень дорогие домашние серверы для мобильных пользователей. Основное их отличие – возможность быстрого перемещения вслед за «летучей» рабочей группой или домашним офисом. Поскольку все взаимодействия с ними обеспечиваются по беспроводной связи, то их применение экономит пользователям много времени и средств, которые бы они раньше потратили на прокладку линий связи. Домашний сервер бе-

рет на себя организацию передачи различных типов информации внутри небольшой локальной компьютерной сети.

По конструктивному исполнению серверы могут быть **башенными, стоечными** и **модульными.**

Модульные серверы

К категории модульных серверов по классификации аналитической компании Gartner относятся блейд-серверы (лезвия) и многоузловые сверхплотные серверные системы. Оба типа модульных серверов используют для размещения стоечные шасси, где помимо самих серверов устанавливаются устройства хранения данных и сетевое оборудование. При этом их можно легко извлекать из шасси и добвалять в него.

Организация блейд-серверов основывается на концепции адаптивной инфраструктуры, которая предусматривает гибкость, экономичность и оперативность подстройки под быстро меняющиеся требования пользователей.

По определению аналитической компании IDC **блейд-сервер** – это модульная одноплатная компьютерная система, включающая процессор и память (рис. 1.9). Лезвия вставляются в специальное шасси (или полку) с объединительной панелью (back plane), обеспечивающей им подключение к сети и подачу электропитания. Это шасси с лезвиями, по мнению IDC, является блэйд-системой. Оно выполнено в конструктиве для установки в стандартную 19-дюймовую стойку и, в зависимости от модели и производителя, занимает в ней 3U, 6U или 10 U (один U – unit, монтажная единица, равная 1,75 дюйма, 1 дюйм равен 2,54 см). За счет общего использования таких компонентов, как источники питания, сетевые карты, жесткие диски и блоки охлаждения, блэйд-серверы обеспечивают более высокую плотность размещения вычислительной мощности в стойке по сравнению с обычными тонкими серверами высотой 1U и 2U. Блейды большинства изготовителей монтируются в шасси вертикально.

К преимуществам использования блейд-серверов можно отнести следующие:

- уменьшение занимаемого объема;
- уменьшение энергопотребления и выделяемого тепла;
- уменьшение стоимости и повышение надежности системы питания и охлаждения;
- повышение удобства управления системой;
- высокая масштабируемость;

- высокая гибкость;
- сокращение количества коммутационных проводов.

Рисунок 1.9. Общая конструкция блейд-сервера

Согласно данным Gartner, в 2011 году мировой рынок блейд-серверов зашел в тупик, так как увеличилась продажа многоузловых серверов. В штучном выражении доля блейдов на мировом рынке серверов в 2012 году составила 12%, что не многим отличается от показателя 2011 года – 13%, в денежном выражении – 21%.

Таким образом, в сегменте блейд-северов фактически ничего не изменилось с 2011 года. Напротив, сегмент многоузловых серверов в 2012 году увеличил долю на рынке с 7,8% до 15% - практически вдвое. Доля этого сегмента в денежном выражении в период с 2011 по 2012 годы также выросла с 5,3% до 10%.

Поскольку конструкция многоузловых серверных систем обеспечивает высокую плотность компонентов и мощные возможности ПО управления серверной инфраструктурой, то их применение в дата-центрах улучшает системное администрирование, эффективность рабо-

ты и экономит пространство, а также обеспечивает гибкость использования разных типов процессоров и поддержку рабочих нагрузок Web-scale. Web-scale IT – это подход к построению архитектуры современных центров обработки данных (ЦОД). Этот подход принят и опробован такими гигантами IT, как Amazon, Facebook, Netflix, Rackspace и Google.

Также Gartner отмечает постепенное стирание границ между лезвиями, многоузловыми серверами, вычислительными модулями для суперкомпьютерных кластеров и серверами для сервис-провайдеров и считает, что теперь все они попадают в категорию модульных серверов.

По данным Gartner, в 2014 году на долю модульных серверов приходилось 26% серверного рынка и их продажи равнялись 13,3 млрд долл., что на 3,4% больше, чем в 2013 году. По прогнозам аналитического агентства, к 2018 году сектор модульных серверов вырастет до 15,5 млрд долл. (годовые темпы роста будут на уровне 3%), что составит около 29% от всего серверного рынка.

Основными покупателями модульных серверов являются крупные корпорации и сервис-провайдеры. Первые из них ценят в модульных серверах высокую управляемость, плотность размещения и быстрое развертывание, вторые — возможность сокращения расходов на серверную инфраструктуру и повышение ее гибкости.

К лидерам рынка модульных серверов Gartner относит HP, Dell и Lenovo, т.е. всю первую тройку крупнейших производителей серверов стандартной архитектуры. Преимущества HP аналитики Gartner видят в ее сильных позициях в секторе блейд-серверов, мощном портфеле многоузловых серверов и наличии в продуктовом портфеле компании модульных серверов для бизнес-критичных приложений Superdome X и NonStop. Однако Gartnet считает, что ее лезвия BladeSystem c-Class, выпущенные в 2006 году, проигрывают более новым серверам Cisco и Lenovo. Кроме того, разнообразие серверного портфеля HP усложняет заказчикам выбор продуктов для дата-центров.

Последние приобретения и новые партнерские соглашения Dell усилили позиции этого вендора, а его система PowerEdge FX2 стала одним из первых модульных серверов для нагрузок Web-scale. К слабым сторонам этого вендора относятся отсутствие четкого плана интеграции продуктового портфеля и недовольство заказчиков качеством услуг препродажных консультаций и поддержки, а также снижение прозрачности бизнеса после преобразования Dell в частную компанию.

Lenovo получила в результате покупки отделения серверов стандартной архитектуры IBM мощный портфель модульных серверов PureSystems, Flex System и NeXtScale, однако вместе с ними и устарев-

шие системы BladeCenter и iDataPlex, которые инсталлированы у многих бывших клиентов IBM и поэтому нуждаются в поддержке. Также многие заказчики не воспринимают китайскую компанию как поставщика сложных высокотехнологичных серверов, и она пока не обнародовала планы дальнейшего развития наследства IBM.

Корпорация Cisco, попавшая в сектор «претендентов», смогла за шесть лет довести продажи блейд-серверов до 3 млрд. долл., однако она проигрывает другим вендорами по разнообразию портфеля модульных серверов. Компания Fujitsu, наряду с Hitachi и Huawei, попавшая в число нишевых игроков, имеет сильные позиции на серверных рынках Германии и Японии, но в других странах она не входит в число лидеров рынка. И, хотя она успешно развивает направление блейд-серверов, ее доля в секторе многоузловых систем остается незначительной.

Эксперты Gartner указывают, что, поскольку многоузловые серверы постепенно захватывают сегмент низко нагруженных систем, то все чаще блейд-серверы используются для комплексных приложений, таких как высоко нагруженные системы, хранилища данных, ERP-системы и т.д.

В исследовании говорится, что большинство блейд-серверов и многоузловых серверов поставляется в x86 архитектуре. Тем не менее, такие вендоры как HP, IBM и Oracle поставляют на рынке и не-x86 блейды, в основном предназначенные пользователям Unix. Такие вендоры как Supermicro, SeaMicro (приобретена AMD), Dell, Huawei и HP производят маломощные серверы на процессорах Intel, AMD и ARM.

Последние новинки блэйд-серверов включают в себя «Storage- Blade» (блейд-системы хранения данных), «Tape-Blade» (блейд-устройства резервного копирования на магнитной ленте), «PCI- Blade» (лезвия для установки полноразмерных плат с интерфейсами PCI, PCI-x или PCI-e).

Рабочие станции

Рабочая станция (Work station), по определению экспертов IDC (International Data Corporation), – это однопользовательская система с мощным одним или несколькими процессорами и многозадачной ОС, имеющая развитую графику с высоким разрешением, большую дисковую и оперативную память и встроенные сетевые средства.

Изначально рабочие станции (WS) ориентируются на **профессиональных пользователей**. Этот вид ЭВМ появился на компьютерном рынке почти одновременно с персональными компьютерами (ПК) и в целом опережает их по своим вычислительным возможностям. В отличие от ПК, ориентированных на самый широкий круг пользователей,

рабочие станции предназначены для корпоративного сектора рынка. Ориентация на корпоративное использование и на профессионального пользователя позволяет во многих случаях применять более совершенные и дорогостоящие аппаратные средства.

Рабочие станции, используя те же процессоры и практически не отличаясь от ПК по внешнему виду, **обладают рядом специфических характеристик**, не свойственных ПК, таких, как поддержка профессиональной двух- и трехмерной графики и многодисковых конфигураций, большой объем и быстродействие жесткого диска, использование двух процессоров (в старших моделях), применение памяти с коррекцией ошибок. Благодаря этому у них выше производительность, надежность и больше графических возможностей, чем у ПК.

Современная рабочая станция – это не просто большая вычислительная мощность. Это тщательно сбалансированные возможности всех подсистем машины, чтобы ни одна из них не стала «бутылочным горлышком», сводя на нет преимущества других. Кроме того, каждая WS, как правило, предназначена для решения определенного класса задач, поэтому в ней используется наиболее эффективное для этого класса аппаратное и программное оснащение.

Традиционными областями применений рабочих станций является работа с компьютерной графикой (трехмерная анимация, создание трехмерных моделей, визуализация различных процессов), автоматизированное проектирование, издательская деятельность. Также WS применяются для осуществления сложных расчетов в самых различных областях науки, при моделировании различных процессов. В этом качестве WS вытеснили с рынка дорогостоящие миниЭВМ, которые как класс компьютеров прекратили свое существование.

В настоящее время рынок рабочих станций существует параллельно с рынком ПК и в ближайшее время будет существовать достаточно независимо. Однако в архитектурном плане рабочие станции и ПК становятся все более схожими. После почти двухлетнего роста мировой рынок рабочих станций сократился. В тройке крупнейших производителей этих устройств изменений не произошло.

Согласно оценкам аналитиков International Data Corporation (IDC), в январе–марте 2015 года поставки рабочих станций в глобальном масштабе составили 838,4 тыс. единиц, что на 1,7 % меньше, чем годом ранее. Эксперты прогнозировали рост на 2,4 %. До минувшей четверти рынок рабочих станций неуклонно рос в течение семи кварталов.

Самый большой прирост объёма отгрузок в первом квартале 2015 года зафиксирован в странах Ближнего Востока и Африки — 11 %, хотя

по итогам последних трёх месяцев прошлого года этот регион показал спад на 15,2 %. В Азиатско-Тихоокеанском регионе (исключая Японию) выпуск рабочих станций поднялся на 10,3 %, здесь превышающие 10 % темпы роста остаются на протяжении шести кварталов кряду.

Крупнейшим производителем рабочих станций в мире по-прежнему является HP, чья рыночная доля в январе–марте выросла до 46,6 %, хотя компания сократила поставки этой продукции — третий раз подряд в квартальном выражении.

Второе место в списке вендоров сохраняет Dell. У неё отгрузки рабочих станций упали в первом квартале на 6,8 % (спад был и кварталом ранее), а рыночная доля — на 0,9 процентного пункта до 35,3 %. Прежде этот показатель у Dell рос в течение семи кварталов.

В отличие от HP и Dell, замыкающая ТОП-3 компания Lenovo смогла нарастить поставки рабочих станций, благодаря чему доля китайского производителя на рассматриваемом рынке поднялась на 2,1 процентного пункта, составив 11,2 %.

На сегодняшний день большинство WS базируется на архитектуре x86 с 4, 6 и 8-ядерными процессорами Intel Xeon, AMD Opteron.

Полностью переосмыслив концепцию рабочих станций, компания HP предложила пользователям решения, которые отличаются беспрецедентной производительностью, функциональностью, удобством обслуживания и выполнены в новом стильном дизайне.

В конструкции рабочих станций HPZ800, Z600 и Z400 Workstations, построенных на базе новой микроархитектуры Intel Nehalem и процессоров Intel Xeon 5500, было реализовано более 20 инноваций HP, в том числе используется блок питания с функцией самопроверки.

Как утверждают представители HP, компьютеры линейки Z стали первыми рабочими станциями HP, которые целиком и полностью – начиная с блока питания и заканчивая материнской платой – могут обслуживаться в буквальном смысле голыми руками, без использования каких-либо инструментов.

В 2014, 2015 годах компания HP представила новые разработки для своих рабочих станций HP Z Workstation, в том числе: систему хранения данных, систему охлаждения и высококачественные дисплеи.

Система хранения данных HP Z Turbo Drive Quad Pro предназначена для увеличения производительности и надёжности крупных хранилищ данных. Рабочие станции Z могут оборудоваться максимум четырьмя сверхбыстрыми модулями HP Z Turbo Drive G2 на одной PCIe x16 карте. Такое решение обеспечивает производительность до 9.0 ГБ/с. Кроме того, HP Z Turbo Drive Quad Pro уникальным образом повышает

целостность данных в случае сбоя в электропитании. HP Z Turbo Drive Quad Pro можно использовать с рабочими станциями HP Z440, Z640 и Z840, разработанными специально для интенсивных вычислений (CAD, архитектура, финансовый сектор, учреждения здравоохранения, обработка изображений, исследование нефти и газа).

HP Z Cooler — инновационная довольно тихая система охлаждения, призванная снизить уровень шума в рабочих станциях HP Z даже при выполнении очень ресурсоёмких задач. Система позволяет снизить скорость вращения кулеров процессора и уменьшает уровень шума до 8.5 акустических дБ — это на 40% тише, чем могли предложить предшественники. Это стало возможным благодаря уникальной, революционной системе теплоотвода HP, сочетающей технологию 3D испаряющей камеры и инновационного дизайна. HP Z Cooler — это элегантное, простое решение, не требующее наличия насосов и трубопроводов, созданное на основе 30-летних наработок HP. HP Z Cooler представляет собой одно из самых бесшумных решений, позволяя, таким образом, повысить продуктивность работы. Система используется на рабочих станциях HP Z440 и Z840.

Дисплеи HP Z22n и HP Z23n Narrow Bezel IPS обеспечивают отличное качество изображения с минимальными искажениями. Панели практически не имеют рамок с трёх сторон и включают фабричную калибровку цвета. Угол обзора составляет 178 градусов благодаря использованию технологии IPS. Цветовая палитра обеспечивает оптимальное качество изображения.

Важным событием в развитии WS явилась совместная разработка HP и Intel архитектуры IA-64, реализующая концепцию EPIC. Линейка процессоров IA-64 Itanium, Itanium 2 нацелена на использование в рабочих станциях и серверах. Модели zx2000, zx6000 содержат несколько многоядерных процессоров Itanium и поддерживают операционные системы HP-UX, Linux, 64-разрядную версию Windows.

Компании Oracle и IBM продолжают развивать RISC-архитектуру, причем последняя разработка IBM – Power 8. Двенадцатиядерный процессор, каждое ядро которого может одновременно выполнять до восьми потоков команд, выглядит серьезным соперником в сфере высокопроизводительных вычислений.

Одной из последних тенденций является удешевление рабочих станций начального уровня при довольно высоком уровне производительности, что позволяет говорить о появлении рынка компьютеров промежуточного уровня между ПК и рабочими станциями, являющихся компромиссом между ценой и производительностью.

Еще одной тенденцией, которую стоит отметить, является появление мобильных рабочих станций.

В качестве примера рассмотрим рабочую станцию Dell Precision M3800. Компания Dell обновила в 2015 г. свой ноутбук Precision M3800, который теперь она называет самой тонкой и легкой в мире 15-дюймовой мобильной рабочей станцией. Эта модель получила мультисенсорный Ultra HD-дисплей и поддержку технологии Thunderbolt 2. Новый дисплей использует технологию IGZO2 и закрыт сверху защитным стеклом Corning Gorilla Glass. Dell утверждает, что Precision M3800 стала первой в отрасли 15-дюймовой рабочей станцией с 4K-дисплеем. Благодаря порту Thunderbolt 2 пользователи получают возможность передавать данные на скорости до 20 Гбит/с. Также Dell увеличила максимальный объём дискового пространства до 2 Тбайт. В компьютере использован процессор Intel Core i7-4702HQ четвертого поколения с 4 ядрами и поддержкой технологии Hyper-Threading. Штатная частота 2.2 ГГц, под нагрузкой до 3.2 ГГц. Под стать процессору и графический чип — Nvidia Quadro K1100M с двумя гигабайтами собственной памяти типа GDDR5. Объем ОЗУ составляет от 8 до 16 Гбайт. По умолчанию на Precision M3800 устанавливается ОС Windows 8.1 Professional. При всех изменениях ноутбук сохранил массу и габаритные размеры предшественника (около 1,88 кг и толщина 18 мм). Стартовая цена новинки составляет $1720 (доплата за 4K-панель составляет $70).

Персональные компьютеры

Персональные компьютеры (ПК) – это однопользовательские микроЭВМ, удовлетворяющие требованиям общедоступности и универсальности применения.

Для удовлетворения этим требованиям персональный компьютер **должен иметь следующие характеристики:**

- невысокую стоимость, находящуюся в пределах доступности для индивидуального покупателя;
- простоту использования;
- возможность индивидуального взаимодействия пользователя с компьютером без посредников и ограничений;
- высокие возможности по переработке, хранению и выдаче информации;
- гибкость архитектуры, обеспечивающую ее адаптивность к разнообразным применениям в сфере управления, науки, образования, в быту;
- высокую надежность, простоту ремонта и эксплуатации;

- «дружественность» операционной системы;
- наличие программного обеспечения, охватывающего практически все сферы человеческой деятельности.

На сегодняшний день **большинство настольных ПК базируется на x86 процессорах** с основной операционной системой из семейства Windows. Эта платформа процессоров является де-факто самой распространенной, демократичной, во многих своих ипостасях гораздо более дешевой и универсальной. Данное направление имеет большое количество клонов, т.е. аналогичных компьютеров, выпускаемых различными фирмами США, Западной Европы, России, Японии и др.

Другая платформа представлена довольно популярными на Западе компьютерами **Macintosh фирмы Apple.** Они занимают на мировом рынке компьютеров довольно узкую нишу, тем не менее удерживают ее за собой в течение уже очень большого промежутка времени с переменным успехом, но все же достаточно стабильно.

Программное обеспечение, операционная система, даже «идеология использования компьютера» – все это в указанных платформах очень сильно разнится. На сегодняшний день формальным отличием является только операционная система (MacOSX, имеющая Unix-подобное ядро). С недавнего времени в компьютерах Macintosh используются процессоры x86 вместо RISC-процессоров Power PC.

По функциональным возможностям и цене персональные компьютеры разделяются на **бюджетные, среднего класса, бизнес-ПК**.

По назначению ПК можно классифицировать: на **бытовые**, **общего назначения**, **профессиональные** и **игровые**.

Бытовые ПК предназначены для массового потребителя, поэтому они должны быть достаточно дешевыми, надежными и иметь, как правило, простейшую базовую конфигурацию. Бытовые ПК используются для обучения, развлечений (видеоигры), управления бытовой техникой, индивидуальной обработки текста, решения небольших инженерных и научных задач, работы в глобальной сети Интернет, хранения видеоинформации и т.д. Бытовой ПК способствовал взрывообразному росту интереса к Интернету, позволив тем самым развить наше представление о мире и сделать его более системным и детальным. По оценке представителей корпорации Intel в современном мире около миллиарда настольных ПК подключены к Интернету. Сегодня Интернет начинает всерьез конкурировать с иными носителями информации, например, 68 % взрослых пользователей Глобальной сети в США предпочитают узнавать новости из Интернета, а не из газет. А семейные фотоальбомы,

столь популярные в России, все увереннее перекочевывают на жесткие диски – это и удобнее и надежнее.

Персональные компьютеры общего назначения применяются для решения различных задач научно-технического и экономического характера, а так же для обучения. Они размещаются на рабочих местах пользователей: на предприятиях, в учреждениях, магазинах, на складах, в вузах, офисах и т.д. Машины этого класса обладают достаточно большой емкостью оперативной памяти. Интерфейсы позволяют подключать большое количество периферийных устройств и средства для работы в составе вычислительных сетей. Минимизированы требования к средствам воспроизведения графики, а к средствам для работы со звуковыми данными требования вообще не предъявляются.

ПК общего назначения используются прежде всего потребителями-непрофессионалами. Поэтому они снабжаются развитым программным обеспечением. Этот класс ПК получил наибольшее распространение на мировом рынке.

Профессиональные ПК используются в научной сфере, для решения сложных информационных и производственных задач, где требуется высокое быстродействие, эффективная передача больших массивов информации, достаточно большая емкость оперативной памяти. Потребителями ПК этого класса, как правило, являются профессионалы-программисты, поэтому программное обеспечение должно быть достаточно богатым, гибким, включать различные программные инструментальные средства. По своим функциональным возможностям профессиональные ПК не только приближаются, но и вполне могут конкурировать с рабочими станциями начального уровня.

Игровые ПК предназначены для компьютерных игр. Основными отличиями игрового ПК являются: производительный процессор, мощная видеокарта, повышенные требования к средствам воспроизведения звука, что обеспечивает достаточно комфортные условия для игры в современные ресурсоемкие компьютерные игры. Благодаря игровым ПК игры стали настоящим искусством. Сорок пять миллионов американцев, или 31 % пользователей Интернета в США, играют в онлайновые игры, всего же в мире более 300 млн геймеров. Большинство из них играет от случая к случаю, а вот 10–15 млн энтузиастов относятся к этому занятию как к главному делу в своей жизни. Не отстает от остального мира и Россия: в течение трех–четырех лет объем российского игрового компьютерного рынка стабильно увеличивался, примерно на 20–25 % в год.

В данном классе ПК необходимо сказать об игровых приставках. **Игровая приставка** (для карманных систем – игровая консоль) – специализированное электронное устройство, разработанное и созданное

для видеоигр. Приставкой устройство называется потому, что оно приставляется к независимому устройству отображения (бытовому телевизору или монитору компьютера). Портативные (карманные) игровые системы имеют собственное встроенное устройство отображения, поэтому называть их игровыми приставками несколько некорректно. По мере развития игровых приставок разница между ними и персональными компьютерами стала постепенно размываться: современные приставки могут позволить подключение клавиатуры, жесткого диска и даже запуск на них операционной системы GNU/Linux. Игровые приставки превратились в наши дни в мощные многофункциональные игровые системы. Например, PlayStation 3 (PS3) – игровая приставка седьмого поколения, третья в семействе игровых систем «PlayStation» корпорации Sony. С помощью PS3 можно играть, смотреть фильмы, слушать музыку, отправлять почту и открывать веб-страницы. Главными конкурентами PS3 являются Xbox 360 от Microsoft и Wii от Nintendo. Пиковая производительность PS3 составляет 2 TFLOPS. Используется восьмиядерный процессор Cell Broadbond Engine с тактовой частотой 3,2 ГГц, совместно разработанный компаниями IBM, Sony и Toshiba. Приставка имеет размеры 325×98×274 мм, вес – 5 кг.

По способу использования ПК можно разделить на два основных класса: **стационарные** (настольные) и **переносные** (мобильные) ПК (рис. 1.12).

В классе **настольных ПК** (desktop) можно выделить **компактные и экологичные десктопы и неттопы** (nettop). В последнее время в корпоративном мире принято считать, что чем меньше размер компьютера, тем лучше. В результате сокращения бюджетов предприятий и уплотнения рабочих мест уменьшается численность IT-персонала и размеры служебных помещений. Все это порождает спрос на более компактные и лучше управляемые, чем стандартные, персональные компьютеры. Такие модели обычно предназначаются для предприятий, но могут быть полезны и в домашнем или малом офисе при дефиците пространства. В качестве примера можно привести компактный настольный ПК Lenovo Think Centre M57/M57P Eco. Китайская компания Lenovo добавила в обозначение компьютеров слово «Eco», подчеркивая минимальное влияние своих изделий на экологическую обстановку. Эти компьютеры стали первыми в мире настольными ПК, получившими сертификацию GREENGUARD, означающую, что ПК были проверены на наличие химических выбросов (до 2000 различных веществ). Данные компьютеры характеризуются компактным размером и по внешнему виду напоминают толстую книгу. Еще одним достоинством

Think Centre M57/M57P Eco является низкий уровень шума. Кроме того, в компьютерах используются технологии, упрощающие эксплуатацию изделия и повышающие безопасность данных.

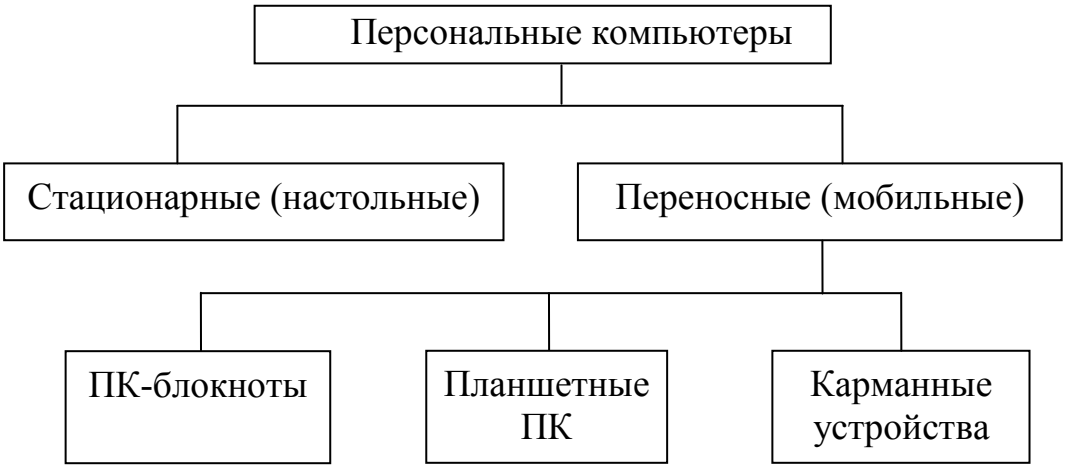


Рис. 1.12. Классификация персональных компьютеров по способу использования

Неттопы – это ориентированные на работу в Интернете настольные ПК на базе процессоров Intel Atom; представляют собой простые в использовании и компактные устройства, имеющие оптимальную производительность для использования всех технологий Интернета. Они отличаются надежностью и гибкими возможностями беспроводной связи. Эти устройства предназначены для обучения, просмотра видео и фотографий, общения в социальных сетях, Интернет-телефонии, работы с электронной почтой, обмена сообщениями, просмотра сайтов и решения других задач.

Мировой рынок настольных ПК в последние годы переживает острейший кризис из-за спада спроса на свою продукцию. Классические десктопы в домашних условиях активно вытесняются, с одной стороны, неттопами, а с другой – стремительно совершенствующимися игровыми приставками; в офисах все чаще устанавливаются «тонкие клиенты».

Тонкий клиент (thin client) в компьютерных технологиях – это бездисковый компьютер-клиент в сетях с клиент-серверной или терминальной архитектурой, который переносит все или большую часть задач по обработке информации на сервер. Тонкий клиент в большинстве случаев обладает минимальной аппаратной конфигурацией.

Увеличивающийся спрос на мобильные ПК объясняется ростом их производительности и снижением цены.

Производители компьютерной техники быстро переориентировались на производство мобильных ПК. В 2009 г. мировой объем поставок ноутбуков превысил объем поставок настольных ПК.

Требования к переносным компьютерам сильно отличаются от требований к настольным ПК. **Мобильные ПК** должны иметь:

- миниатюрные внутренние компоненты и периферийные устройства;
- автономное электропитание;
- низкое энергопотребление;
- малые габариты и вес.

Переносные ПК по своим конструктивным особенностям можно разделить: на ПК-блокноты (ноутбуки), планшетные ПК и карманные устройства. Существует термин **лэптоп** (laptop – наколенный), который применяется как к ноутбукам, так и к планшетным ПК. К ноутбукам обычно относят лэптопы, выполненные в раскладном форм-факторе.

ПК-блокноты (ноутбуки)

Все **ноутбуки** (notebook) классифицируются на несколько типовых разновидностей по размеру диагонали дисплея, назначению, компоновке составных узлов, функциональным возможностям, габаритам, весу и другим отличиям. К основным типам ноутбуков можно отнести: «замену настольного ПК» (Desktop Replacement), массовые ноутбуки, ультрабуки, смартбуки.

В качестве **замены настольного ПК** обычно позиционируются ноутбуки с диагональю экрана 17 дюймов и выше. Габариты и вес (от 3 кг и выше) портативных компьютеров весьма значительны, что делает их неудобными в переноске. Однако относительно большой размер дисплея обеспечивает более комфортную работу, а объемистый корпус позволяет установить мощные компоненты и обеспечить им достаточное охлаждение. Такие ноутбуки имеют встроенные жесткий диск, аккумулятор, CD или DVD-привод, порты ввода/вывода. Снаружи подсоединяется блок питания, как у всех других ноутбуков. Одним из самых мощных и дорогих ноутбуков категории Desktop Replacement в 2015 г. является ASUS ROG G751JL с размером экрана по диагонали 17,3', с разрешение 1920x1080 точек. Процессор – Intel Core i7-4720HQ с частотой 2,6 ГГц. Оперативная память до 32 Гбайт, видеокарта – NVIDIA GeForce GTX 965M с двумя гигабайтами собственной памяти. Вес – 4,5 кг. Стоимость $2500.

Массовые ноутбуки (специального названия для данной категории ноутбуков не предусмотрено) имеют диагональ экрана 14'-16', их вес обычно укладывается в 2–3 кг, толщина оказывается чуть меньше ноут-

буков «замена настольного ПК». Обычно эти модели оснащены встроенными жестким диском и оптическим накопителем.

Ультрабуки (ultrabooks) – тонкий и легкий ноутбук, обладающий ещё меньшими габаритами и весом по сравнению с обычными ноутбуками, но при этом – большей частью характерных черт полноценного ноутбука. Термин стал широко распространяться в 2011 году, после того как корпорация Intel презентовала новый класс мобильных ПК – ультрабуки.

Немного истории. Первоначально концепция мобильного компьютера, более компактного и лёгкого, чем обычный ноутбук, появилась в 1996 году, когда корпорация Toshiba выпустила семейство ноутбуков Toshiba Libretto. Этот класс компьютеров получил наименование субноутбуки. С тех пор в течение 15 лет субноутбуки постоянно развивались в направлении снижения габаритов и цены и увеличения вычислительной мощности и длительности автономной работы от встроенной аккумуляторной батареи.

15 января 2008 года Стив Джобс провёл презентацию нового сверхлёгкого субноутбука Apple MacBook Air, выполненного в сверхтонком алюминиевом корпусе и не имевшего аналогов на тот момент. После начала продаж выяснилось, что данный субноутбук имеет повышенный спрос у потребителей, и вскоре стали появляться аналоги от других производителей ноутбуков: Dell Adamo, Lenovo ThinkPad X300, Samsung 900X3A, Sony Vaio Y.

В мае 2011 года появился новый класс мобильных ПК – ультрабуки, который является дальнейшим эволюционным развитием классических субноутбуков и во многом использует идеи, реализованные в сверхтонком ноутбуке от Apple, MacBook Air.

Нетбуки (netbooks) как отдельная категория ноутбуков были выделены из категории субноутбуков в 2008 г. компанией Intel. Размер диагонали экрана нетбуков – от 7' до 12,1'. Нетбуки ориентировались на просмотр веб-страниц, работу с электронной почтой и офисными программами. Для этих ноутбуков были разработаны специальные энергоэффективные процессоры Intel Atom, VIA C7, VIA nano, AMD Geode. Малый размер экрана, небольшая клавиатура и низкая производительность подобных устройств компенсировались умеренной ценой и относительно большим временем автономной работы. Габариты обычно не позволяли устанавливать в нетбук дисковод оптических дисков, однако Wi-Fi-адаптер являлся обязательным компонентом. Столкнувшись с конкуренцией со стороны ультрабуков и планшетных ПК, натиск последних выдержали лишь компнаии Asustek и Acer, которые продавали

свои нетбуки плоть до конца 2012 года в основном на разививающихся рынках Южной Азии и Южной Африки. Эра нетбуков закончилась в 2012 г. В 2013 г. распродавались только их запасы.

В 2015 году компания Microsot неожиданно для многих, кроме планшета Surface Pro 4, представила также ультрабук Surface Book. Сейчас такие устройства принято называть гибридными.

Однако Microsoft называет новинку просто ноутбуком. В этом случае в первую очередь обращает на себя внимание дисплей диагональю 13,5 дюйма. У него крайне необычное для ноутбуков соотношение сторон (3:2) и разрешение (3000 х 2000 точек).

С технической точки зрения аппарат похож на новый планшет Microsoft. Тут используется корпус из того же магниевого сплава, а дисплей также располагает специальным слоем для работы со стилусом. К слову, перо Surface Pen поставляется в комплекте с новинкой.

Необычным выглядит конструкция петель. Несмотря на отключаемую планшетную часть, инженеры Microsoft наделили устройство возможностью раскрыть дисплей на 360°.

Сердцем ноутбука служат процессоры Intel Core i5 или i7 поколения Skylake. В оперативной памяти предусмотрено 8 либо 16 ГБ. Для хранения данных присутствует SSD объёмом 128, 256, 512 ГБ либо 1 ТБ. Что любопытно, в продаже будут модификации ноутбука с дискретными видеокартами Nvidia. Модели не называются, но данный компонент расположен в клавиатурном блоке. Ёмкости аккумулятора должно быть достаточно для 12 часов в режиме проигрывания видео.

Габариты ноутбука составляют 312,3 х 232,1 х 13-22,8 мм при массе 1,5 кг с подключенной клавиатурой. Список портов представлен парой USB 3.0, Mini DisplayPort и слотом для карт SD. В минимальной конфигурации ноутбук обойдётся покупателям в $1500, а за версию с процессором Core i7 и видеокартой Nvidia придётся отдать $2700.

В 2009 г. разработчики и производители компьютерной техники заговорили о новой категории компьютеров под названием смартбуки.

Смартбук – это небольшой компьютер с дисплеем и клавиатурой, представляющий собой нечто среднее между смартфоном и нетбуком. По размерам он меньше нетбука, а по функциональным возможностям аналогичен смартфону. Смартбук способен обеспечивать постоянное беспроводное 3G-соединение и работать не менее 8 часов без подзарядки. Он обладает экраном с диагональю от 7 до 9 дюймов и может базироваться на процессорах с архитектурой ARM под управлением ОС на ядре Linux, например Google Android.

Статистика использования в настольных ПК и ноутбуках различных ОС на декабрь 2015 г. представлена в таблице 1.2:

Таблица 1.2

Статистика использования ОС на декабрь 2015 г.

Название ОС	**Использование, %**
Windows 7	49.27
Windows 8.1	13.02
Windows 10	10.18
OS X	9.36
Windows XP	8.52
Windows 8	2.96
Windows Vista	1.74
Linux	1.50
Other/Unknown	3.47

Среди корпоративных клиентов Windows 10 получила массу положительных откликов, однако в четвертом квартале 2015 года компании лишь тестировали эту ОС и не торопились переходить на нее.

В январе 2016 года аналитики IDC сообщили об обвале мирового рынка персональных компьютеров. Его объем в 2015 году впервые за семь лет оказался ниже 300 млн. единиц, а спад — рекордным.

Согласно расчетам экспертов, в 2015 году вендоры отгрузили по всему миру 276,2 млн. десктопов и ноутбуков, что на 10,4% меньше, чем годом ранее. Столь сильного падения на рынке не было никогда. Прежде сильнейший регресс (на 9,8%) произошел в 2013 году, а поставки ПК последний раз падали ниже 300 млн. устройств лишь в 2008 году, отмечается в исследовании.

Падение компьютерной отрасли эксперты объясняют рядом причином: увеличенным сроком использования ПК, конкуренцией со стороны смартфонов и планшетов, экономическими потрясениями (девальвация мировых валют и падение цен на нефть) и сильным 2014 годом, когда спрос стимулировался прекращением поддержки ОС Windows XP и продвижением недорогих устройств.

В пятерке ведущих ПК-вендоров в 2015 году произошло лишь одно изменение: Apple вышла на четвертое место, опередив Acer и ASUS. Корпорация Apple осталась единственным крупнейшим производителем с растущими поставками компьютеров. Более подробно о расстановке сил среди брендов можно узнать из таблицы 1.3.

Таблица 1.3

Топ-5 крупнейших производителей ПК, 2015 г

Вендор	Поставки в 2015 г	Доля в 2015 г	Поставки в 2014 г	Доля в 2014 г	Рост в 2015 г
1. Lenovo	57,182	20,7%	59,306	19,2%	**-3,6%**
2. HP	53,534	19,4%	56,869	18,4%	**-5,9%**
3. Dell	39,049	14,1%	41,509	13,5%	**-5,9%**
4. Apple	20,794	7,5%	19,575	6,3%	**6,2%**
5. Acer	19,680	7,1%	24,043	7,8%	**-18,1%**
Другие	85,977	31,1%	107,063	34,7%	**-19,7%**
Итог	**276,216**	**100,0%**	**308,365**	**100,0%**	**-10,4%**

В 2015 году одной из главных проблем для компьютерной отрасли была девальвация мировых валют. Именно из-за этого фактора выпуск ПК в Японии и странах Европы, Ближнего Востока, Африки и Латинской Америки по итогам 2015 года уменьшился почти на 10% в сравнении с 2014-м. В США и Азиатско-Тихоокеанском регионе регресс был минимальным, поскольку колебания курсов валют не повлияли на эти рынки.

В октябре–декабре 2015 года объем глобальной компьютерной индустрии снизился на 8,3% в сравнении с аналогичным периодом предыдущего года, составив 75,7 млн штук. Этот квартальный спад оказался пятым подряд и показал то, что даже предновогодний сезон не способствует общему наращиванию спроса на ПК, что в свою очередь указывает на изменения покупательского поведения потребителей в отношении этих устройств.

Планшетные ПК

Планшетные ПК (Tablet PC) – устройства с жидкокристаллическими сенсорными дисплеями, позволяющими работать без использования клавиатуры. Корпорация Microsoft предложила новый тип мобильных компьютеров еще в 2002 г. Отдельные технологические решения, входящие в состав платформы, давно известны. Однако корпорация Microsoft, объединив достижения ноутбукостроения, оцифровки графики, беспроводной связи и усовершенствовав программы для распознавания рукописного текста и голоса, сумела создать новую, логически завершенную концепцию работы с информацией в «полевых» условиях.

В планшетный компьютер можно ввести данные, «написав» их специальным пером прямо на поверхности монитора, причем не стараясь выводить печатные символы, а просто так, как вы делали бы это в бумажном блокноте. Более того, информацию можно просто надикто-

вать в микрофон планшетного компьютера, а соответствующая программа переведет речь в обычный текст. Беспроводный доступ к локальным компьютерным сетям поможет тут же передать информацию по назначению или запросить нужные данные, в том числе из сети Интернет. Конструктивно планшетный компьютер – это дисплей, под которым спрятана элементная база обычного современного ноутбука: процессор, жесткий диск, оперативная память и модули беспроводного доступа. Некоторые модели снабжены собственной клавиатурой. Планшетные компьютеры более легкие и мобильные, чем обычные ноутбуки. Они обладают отличными демонстрационными возможностями.

Статистика продаж на мировом рынке планшетных ПК в 2014, 2015 г. представлена в таблице 1.4 (млн. шт.).

Таблица 1.4

Статистика продаж на мировом рынке планшетных ПК

Вендор	**Поставки в 2015 г.**	**Доля в 2015 г.**	**Поставки в 2014 г.**	**Доля в 2014 г.**	**Рост от года к году**
1. Apple	49,6	24,0%	63,4	27,6%	-21,8%
2. Samsung	33,4	16,2%	39,8	17,3%	-16,1%
3. Lenovo	11,2	5,4%	11,2	4,9%	0,4%
4. ASUS	7,1	3,4%	11,8	5,1%	-39,9%
5. Huawei	6,5	3,1%	3,0	1,3%	116,6%
Другие	99,1	47,9%	100,9	43,8%	-1,8%
Итог	**206,8**	**100,0%**	**230,1**	**100,0%**	**-10,1%**

В качестве примера рассмотрим планшетные компьютеры корпорации Apple.

За четыре года существования из красивой игрушки планшеты Apple iPad превратились в серьёзных конкурентов ноутбукам - именно этим лёгким и мобильным гаджетам люди всё чаще отдают предпочтение. Apple умеет удивлять, и очередное поколение iPad не стало исключением из правил. Apple iPad Air 2 ещё более мощный и производительный, но вместе с тем ещё более тонкий и лёгкий планшет.

Его характеристики:

- экран: 9,7', 1536x2048, IPS, олеофобное покрытие, ламинированный дисплей;
- процессор: Apple A8X, 3 ядра, 1,5 ГГц, 64 бит;
- графический ускоритель: PowerVR GX6650;
- операционная система: iOS 8;
- оперативная память: 2 ГБ;
- встроенная память: 16/64/128 ГБ;
- камеры: основная — 8 Мп, фронтальная — 1,2 Мп;

- датчики: сканер отпечатков пальцев Touch ID, трёхосевой гироскоп, акселерометр, датчик освещённости, барометр;
- габариты: 240 x 170 x 6,1 мм;
- вес: 437 граммов.

В планшете стоит сопроцессор, который обрабатывает данные акселерометра, компаса и барометра.

Карманные устройства

Карманные устройства с диагональю экрана менее 7' выделяют в специальную категорию «наладонных компьютеров» (hand held PC), которые можно подразделить на карманные ПК (КПК), мобильные интернет-устройства (MID), смартфоны и коммуникаторы.

Мобильное Интернет-устройство (mobile Internet device, MID) – это карманное абонентское устройство с диагональю экрана от 4 до 7 дюймов, предназначенное для беспроводного доступа в Интернет; как правило, поддерживает технологии Wi-Fi, WiMAX или 3G, работает на легких ОС с очень быстрым запуском, например на Linux. Оптимизировано для получения информации и для веб-серфинга (просмотра сайтов). По размеру MID меньше ноутбука и немного больше смартфона. Это мощные карманные устройства стоимостью 300–400 евро. Среди производителей MID – такие бренды, как Aigo, Asus, Ben Q, Lenovo, Toshiba, Compal и др. Рынок MID в 2013 г. превысил 130 млн штук. Темпы роста рынка будут во многом зависеть от темпов внедрения сетей широкополосного беспроводного доступа и выпуска вычислительных платформ с низким энергопотреблением.

Карманный персональный компьютер (КПК) – портативное вычислительное устройство, обладавшее широкими функциональными возможностями, начиная от чтения электронных книг и кончая выполнением офисных приложений. Изначально КПК предназначались для использования в качестве электронных органайзеров. На английском языке словосочетание «карманный ПК» (**Pocket PC**) являлось **торговой маркой** фирмы Microsoft, т.е. относилось лишь к одной из разновидностей КПК, а не обозначало весь класс устройств. Словосочетание **Palm PC** («наладонный компьютер») также являлось конкретной торговой маркой. Для обозначения всего класса устройств в английском языке использовалась аббревиатура **PDA – Personal Digital Assistant** – «личный цифровой секретарь». К карманному ПК, оснащенному хост-контроллером USB, можно было напрямую подключать различные USB-устройства, в том числе клавиатуру, мышь, жесткий диск и флэш-

накопитель. Основными операционными системами для КПК являлись: Windows Mobile (ранее Pocket PC и Windows CE) фирмы Microsoft; Palm OS фирмы Palm Source. Начиная с 2006 г., объемы поставок КПК стали постоянно снижаться и в 2008 г. КПК были практически вытеснены смартфонами и коммуникаторами.

Смартфоны и коммуникаторы. В настоящее время не существует четкого разграничения между смартфонами и коммуникаторами, поскольку функциональность обоих классов устройств примерно одинакова. Различные эксперты и производители по-разному трактуют эти термины. Часто применяется так называемый «исторический подход», который заключается в следующем: если устройство ведет свою родословную от КПК – это коммуникатор, а если от мобильных телефонов – это смартфон. Таким образом, **смартфон** (smartphone – интеллектуальный телефон) – мобильный телефон с расширенной функциональностью, сравнимой с КПК; **коммуникатор** (communicator, PDA Phone) – карманный персональный компьютер, дополненный функциональностью мобильного телефона. В рамках этого подхода изначально под коммуникаторами подразумевались устройства с сенсорным экраном (мог быть дополнен клавиатурой). Устройства, использовавшие для ввода информации исключительно полноразмерную (QWERTY) клавиатуру и/или цифровую клавиатуру (аналог телефонной), назывались смартфонами. В большинстве случаев позиционирование устройства зависело от производителя (обычно устройства с сенсорным экраном относились к коммуникаторам, а к смартфонам относились устройства без такого экрана). Часть специалистов разделяло коммуникаторы и смартфоны соответственно наличием или отсутствием полноразмерной (QWERTY) клавиатуры (виртуальной или физической). В настоящее время в смартфонах и коммуникаторах используется небольшое количество ОС. По данным компании IDC во втором квартале 2015 г. на рынке доминировала ОС Android (82,8 %), на втором месте – iOS (13,9 %), на третьем – Windows Phone (2,6 %), другие 0,7 %.

В 2012 г. на рынке появлилась новая разновидность смартфонов под названием фаблеты. **Фаблет (phablet)** – это сматрфон, имеющий диагонал экрана от 5,5 дюймов, большие габариты и более долгий срок использования. В четвертом квартале 2014 г. продажи фаблетов почти в два раза превысили продажи планшетов.

Победителем рейтинга лучших смартфонов 2015 г. стала модель Samsung Galaxy S6 Edge (табл. 1.5). Модель может похвастаться первым в мире двойным изогнутым дисплеем – с экраном, который закруглен с обеих сторон, – так что она не похожа ни на один из существующих те-

лефонов. Этот смартфон, в сочетании с наилучшей в данной отрасли мощностью, потрясающим дисплеем, достойным сроком службы батареи, отличной камерой и поразительными эксплуатационными качествами является не только самым красивым смартфоном, но и входит топ-2 лучших смартфонов во всей вселенной.

Таблица 1.5

Характеристики смартфона Samsung Galaxy S6 Edge.

Экран	5.1 дюймов (~71.7% — соотношение экрана к корпусу); Super AMOLED сенсорный, 16M цветов; 1440 x 2560 пикселей (~577 ppi пиксельная плотность)
Процессор	Quad-core 1.5 ГГц Cortex-A53 & Quad-core 2.1 ГГц Cortex-A57
Встроенная память	32/64/128 ГБ, 3 ГБ RAM
Слот под карты памяти	Отсутствует
Камеры	Главная: 16 МП; Фронтальная: 5 МП;
Батарея	Несъемная Li-Ion 2600 mAh
Цена	$ 850.00

Вторую позицию занял фаблет iPhone 6S Plus.

Статистика продаж мирового рынка смартфонов за третий квартал 2014, 2015 г. представлена в таблице 1.6 (млн. шт.).

Таблица 1.6

Статистика продаж мирового рынка смартфонов

Вендор	**Отгрузки в 3 кв. 2015 г.**	**Доля рынка в 3 кв. 2015 г.**	**Отгрузки в 3 кв. 2014 г.**	**Доля рынка в 3 кв. 2014 г.**	**Годовая динамика**
Samsung	84.5	23.8%	79.6	23.9%	6.1%
Apple	48.0	13.5%	39.3	11.8%	22.2%
Huawei	26.5	7.5%	16.5	5.0%	60.9%
Lenovo*	18.8	5.3%	16.9	5.1%	11.1%
Xiaomi	18.3	5.2%	17.3	5.2%	5.6%
Прочие	159.1	44.8%	163.0	49.0%	-2.4%
Всего	**355.2**	**100.0%**	**332.6**	**100.0%**	**6.8%**
Lenovo + Motorola	18.8	5.3%	25.7	7.7%	-26.8%

Встраиваемые и промышленные компьютеры

Встраиваемые микроЭВМ входят составным элементом в промышленные и транспортные системы, технические устройства и аппараты, бытовые приборы. Они способствуют существенному повышению их эффективности функционирования, улучшению технико-экономических и эксплуатационных характеристик.

В области мобильных и малогабаритных аппаратов традиционно применяются специализированные процессоры, такие как RISC-системы с архитектурой ARM, энергопотребление которых находится

на уровне 3 Вт (универсальные микропроцессоры имеют более высокое энергопотребление).

Появление компьютеров в модульном исполнении (Computer-on-Module, COM) на базе Intel Atom с операционной системой Microsoft, ориентированной на встраиваемые системы на платформе x86 (Windows Embedded Standard), может потеснить RISC-системы в этой области.

Компьютерные модули компании Kontron на основе Atom (nano ETX express-SP), мощность которых находится в пределах 5 Вт, хотя и не дотягивают по этому параметру до ARM, становятся уже вполне конкурентноспособными в мобильном сегменте.

Встраиваемые платформы Microsoft на «атомных» компьютерных модулях могут быть использованы при создании различных переносных устройств медицинского или специального назначения, портативной мультимедийной и навигационной техники, компактных систем для работы с различными данными и множества других малогабаритных и мобильных устройств и систем.

Промышленные компьютеры используются как автономные человеко-машинные интерфейсы (Human Machine Interface, HMI) и промышленные терминалы в приложениях с жесткими условиями эксплуатации. К ним предъявляются серьезные требования по защите поверхности и герметичности корпуса. В качестве примера можно привести серию бюджетных безвентиляторных промышленных компьютеров класса Touch Panel PC (компьютеры с сенсорным экраном) под названием Nano Client компании Kontron. Компьютеры этого семейства построены на базе 45нм процессоров Intel Atom Z5xx с низким энергопотреблением. Тактовая частота центрального процессора достигает 1,6 ГГц, объем запаянной памяти (ОЗУ) – до 1024 Мб, что, как утверждают разработчики, позволяет реализовывать на основе этих компьютеров сложные схемы визуализации. Упрощенная схема охлаждения позволила инженерам Kontron создать недорогое оптимизированное решение в полностью герметичном стальном корпусе. Компьютеры Nano Client оснащаются сенсорными экранами диагональю 10,4 или 15,0 дюймов, их толщина составляет всего 63 мм. Стандартный набор поддерживаемых ОС включает Windows CE.NET/XP Embedded и Embedded Linux.

Изделия серии Kontron Nano Client предназначены для самых различных «жестких» приложений – это человеко-машинные интерфейсы, монтируемые на поворотный кронштейн (на станках с ЧПУ, промышленных весах и др.); терминальные тонкие клиенты в таких задачах, как обработка заказов, контроль качества, склад и торговля.

2. ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЭВМ

Существуют два взгляда на построение и функционирование ЭВМ. Первый – взгляд пользователя, не интересующегося технической реализацией ЭВМ и озабоченного только получением некоторого набора функций и услуг, обеспечивающих эффективное решение его задач; второй – разработчика ЭВМ, усилия которого направлены на рациональную техническую реализацию необходимых пользователю функций. С учетом этого обстоятельства и вводятся понятия «функциональная» и «структурная» организация компьютера.

Действительно, с точки зрения пользователя решение любой задачи на ЭВМ требует поэтапного выполнения некоторой последовательности действий: программирования, кодирования, ввода, обработки, документирования. На каждом из этих этапов учет запросов пользователя может потребовать расширения реализуемых ЭВМ функций и услуг, что решается при проектировании ЭВМ и входит в понятие **функциональной организации ЭВМ**.

В результате создается абстрактная модель ЭВМ, описывающая функциональные возможности машины и предоставляемые ею услуги. Функциональная организация ЭВМ в значительной степени определяется предъявляемыми к ней требованиями, уровнем подготовки потенциальных пользователей, типом решаемых ими задач, потребностями в развитии компьютера (по емкости памяти, разрядности, составу периферийных устройств и др.).

Предусматриваемые абстрактной моделью функции ЭВМ реализуются на основе реальных физических средств (устройств, блоков, узлов, элементов) в рамках определенной структуры. В общем случае под **структурной организацией ЭВМ** понимается некоторая физическая модель, устанавливающая состав, порядок и принципы взаимодействия основных функциональных частей машины (без излишних деталей их технической реализации).

Функциональная организация ЭВМ играет ведущую роль и в значительной степени определяет структурную организацию машины, хотя и не дает жестких ограничений на конечную техническую реализацию структурных элементов. Вместе с тем функции и структура любого элемента находятся в диалектической взаимосвязи и взаимозависимости. С одной стороны, функциональным назначением устройства (блока, уз-

ла) ЭВМ определяется необходимый состав материальных объектов (реальных аппаратных и программных средств) и характер их взаимодействия. С другой стороны, одна и та же функция может быть реализована на совершенно разных технических средствах, а изменение состава или связей между элементами, изменение пропорций между аппаратными и программными средствами может сохранить неизменной функцию системы, сообщив ей новые свойства.

2.1. Обобщенная структура ЭВМ и пути её развития

Развитие архитектуры неизбежно ведет к развитию структуры ЭВМ. Реализация принципов интеллектуализации, которые все больше определяют развитие архитектуры ЭВМ, возможна при совершенствовании структурной организации, обеспечивающей повышение эффективности вычислительного процесса и, как следствие этого, рост производительности ЭВМ. В конечном счёте условием и критерием развития структуры является рост производительности ЭВМ.

Основной тенденцией в развитии структуры ЭВМ является разделение функций системы и максимальная специализация подсистем для выполнения этих функций.

Обобщенная структура ЭВМ приведена на рис. 2.1. Она состоит из следующих составных частей: обрабатывающей подсистемы; подсистемы памяти; подсистемы ввода/вывода; подсистемы управления и обслуживания.

Для каждой подсистемы выделены основные направления их развития.

Обрабатывающая подсистема

Развитие обрабатывающей подсистемы в большей степени, чем всех остальных подсистем, идет по пути разделения функций и повышения специализации составляющих ее устройств. Создаются специальные средства, которые осуществляют функции управления системой, освобождая от этих функций средства обработки. Такое распределение функций сокращает эффективное время обработки информации и повышает производительность ЭВМ. В то же время, средства управления, как и средства обработки, становятся более специализированными. Устройство управления памятью реализует эффективные методы передачи данных между средствами обработки и подсистемой памяти. Меняются функции центрального устройства управления.

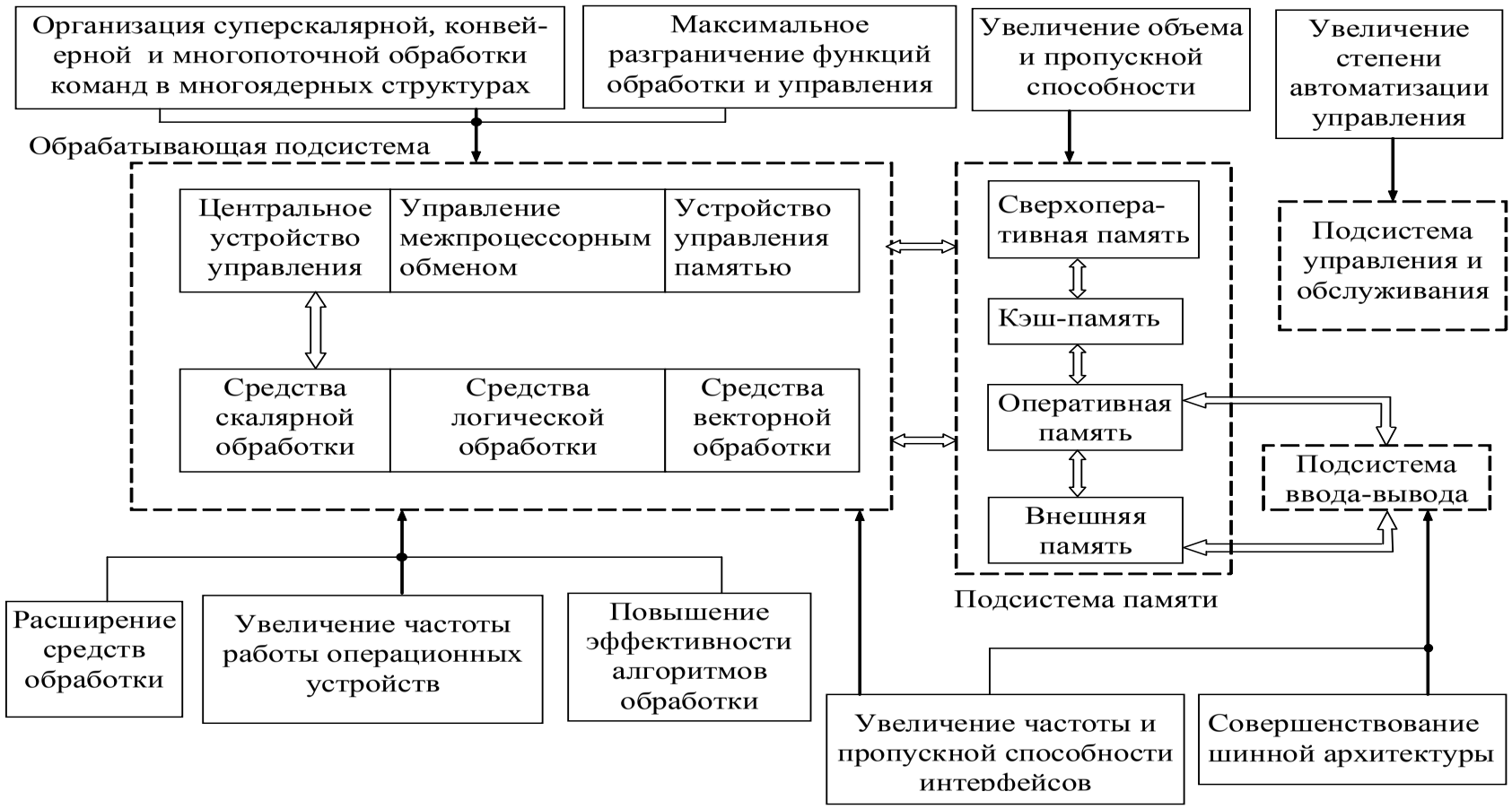


Рис. 2.1. Обобщенная структура ЭВМ и основные направления ее развития

С одной стороны, ряд функций передается в другие подсистемы (например, функции ввода/вывода), с другой – развиваются средства организации параллельной обработки нескольких команд (суперскалярная обработка, конвейерная технология выполнения команд, многоядерные структуры, многопотоковая обработка команд, динамическое изменение последовательности команд, предварительная выборка данных, предсказание направления ветвлений и т.д.). Бурно развивается управление межпроцессорным обменом как эффективное средство передачи информации между несколькими центральными процессорами, входящими в состав вычислительной системы или комплекса.

Операционные устройства обрабатывающей подсистемы, кроме традиционных средств скалярной (суперскалярной) и логической обработки, все шире стали включать специальные средства векторной обработки. При этом время выполнения операций можно резко сократить за счет увеличения частоты работы операционных устройств.

В устройствах скалярной обработки все шире появляются специальные операционные блоки, оптимизированные на эффективное выполнение отдельных операций, разрядность обрабатываемых слов возрастает.

Подсистема памяти

Подсистема памяти современных компьютеров имеет иерархическую структуру, состоящую из нескольких уровней:

- сверхоперативный уровень (память процессора, кэш-память);
- оперативный уровень (оперативная память);
- внешний уровень (внешние ЗУ на дисках, лентах и т.д.).

Эффективными методами повышения производительности ЭВМ являются увеличение количества регистров общего назначения процессора, использование многоуровневой кэш-памяти, увеличение объема и пропускной способности оперативной памяти (ОП), буферизация передачи информации между ОП и внешней памятью.

Подсистема ввода/вывода

В состав подсистемы ввода/вывода входит набор специализированных устройств, между которыми распределены функции ввода/вывода, что позволяет свести к минимуму потери производительности системы при операциях ввода/вывода. Эти устройства определяют пропускную способность подсистемы ввода/вывода.

Основными направлениями развития подсистем ввода/вывода являются совершенствование системных контроллеров и контроллеров

ввода/вывода, увеличение частоты и пропускной способности интерфейсов, совершенствование шинной архитектуры.

Подсистема управления и обслуживания

Подсистема управления и обслуживания – это совокупность аппаратно-программных средств, предназначенных для обеспечения максимальной производительности, заданной надежности, ремонтопригодности, удобства настройки и эксплуатации. Она обеспечивает проблемную ориентацию и заданное время наработки на отказ, подготовку и накопление статистических сведений о загрузке и прохождении вычислительного процесса, выполняет функции «интеллектуального» интерфейса с различными категориями обслуживающего персонала, осуществляет инициализацию, тестирование и отладку. Подсистема управления и обслуживания позволяет поднять на качественно новый уровень эксплуатацию современных ЭВМ.

При разработке структуры ЭВМ все подсистемы должны быть сбалансированы между собой. Только оптимальное согласование быстродействия обрабатывающей подсистемы с объемами и скоростью передачи информации подсистемой памяти, с пропускной способностью подсистемы ввода/вывода позволяет добиться максимальной эффективности использования ЭВМ.

Важнейшими факторами, определяющими функциональную и структурную организацию ЭВМ, являются выбор системы и форматов команд, типов данных и способов адресации.

2.2. Типы данных

Основными скалярными типами данных в компьютерах интеловской архитектуры являются: байт, слово, двойное слово и квадрослово (рис. 2.2).

Основными векторными типами данных являются: 64, 128, 256 и 512-битные вектора, используемые в SIMD командах.

Каждый из представленных на рис. 2.2 скалярных типов данных может начинаться с любого адреса: это означает, что слово не обязано начинаться с чётного адреса; двойное слово – с адреса, кратного 4 и т.д. Таким образом, достигается максимальная гибкость структур данных и эффективность использования памяти.

На базе основных типов данных строятся все остальные типы, распознаваемые командами процессора.

Целочисленные данные

Четыре формата данных (байт, слово, двойное слово, учетверенное слово) с фиксированной точкой могут быть как со знаком, так и без знака. Под знак отводится старший бит формата данных. Представление таких данных и выполнение операций в арифметико-логическом устройстве (ALU) производится в дополнительном коде.

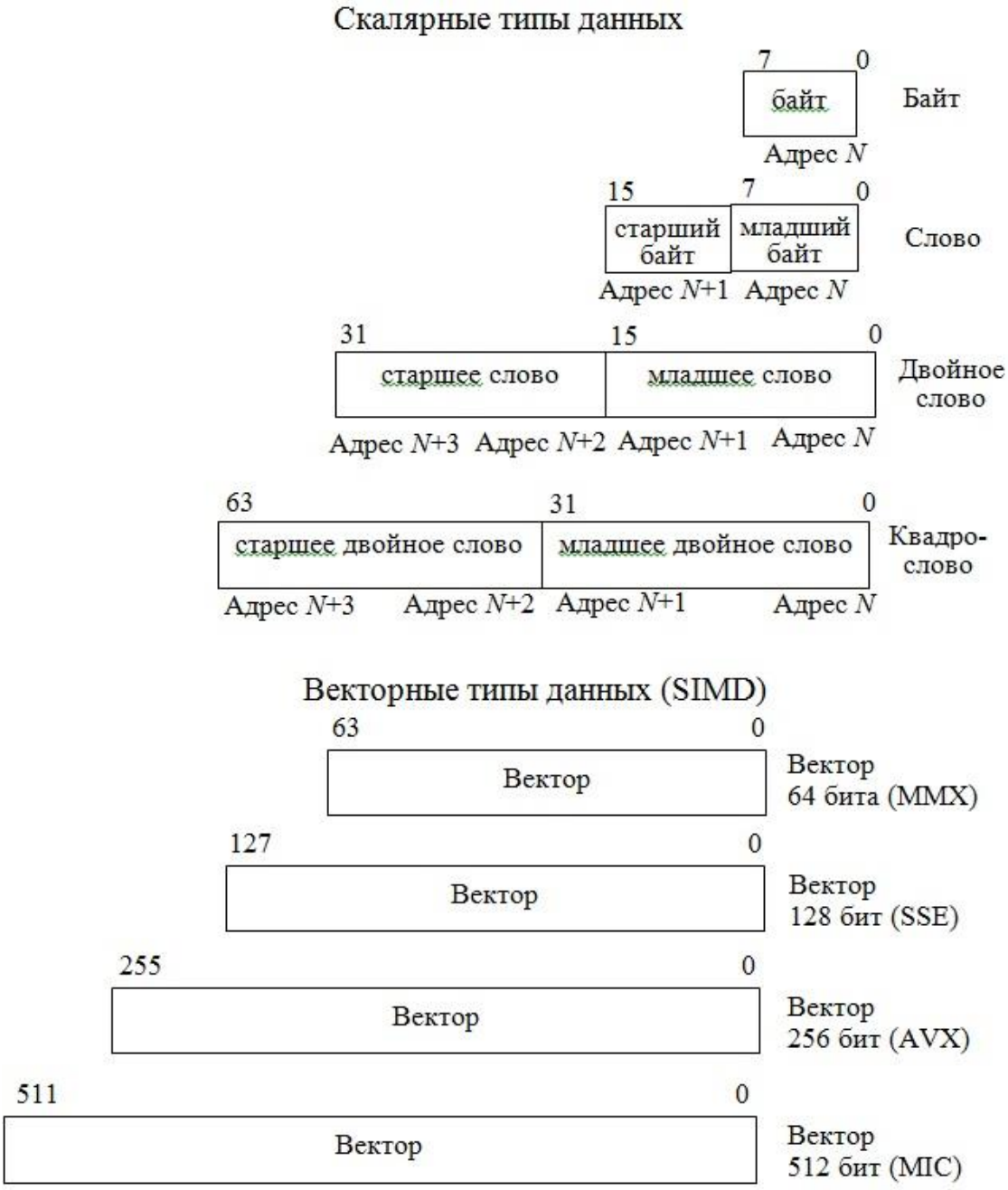


Рис. 2.2. Основные типы данных

Данные в формате с плавающей точкой x87

Формат включает три поля: Знак (*S*), Порядок и Мантисса (рис. 2.3). Поле мантиссы содержит значащие биты числа, а поле порядка содержит степень 2 и определяет масштабирующий множитель для мантиссы. Форматы данных поддерживаются блоком обработки чисел с плавающей точкой (FPU).

Рис. 2.3. Форматы данных с плавающей точкой

Двоично-десятичные данные (BCD)

На рис. 2.4 приведены форматы двоично-десятичных данных.

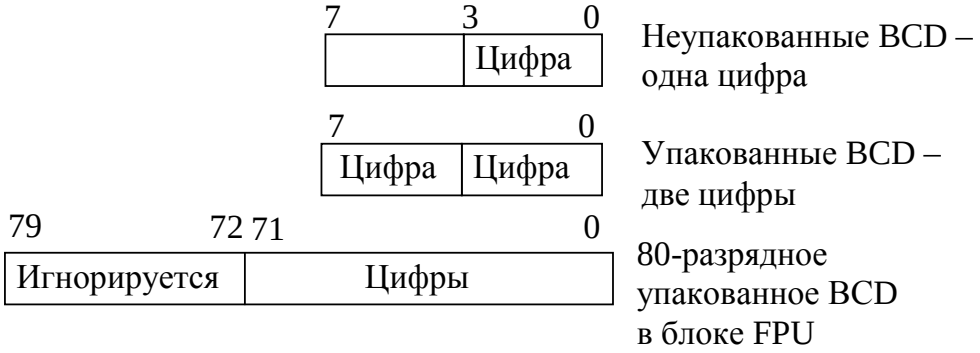


Рис. 2.4. Форматы двоично-десятичных данных

Данные типа строка

Строка представляет собой непрерывную последовательность бит, байт, слов или двойных слов (рис. 2.5). Строка бит может быть длиной до 1 Гбита, а длина остальных строк может составлять от 1 байта до 4 Гбайтов. Поддерживается ALU.

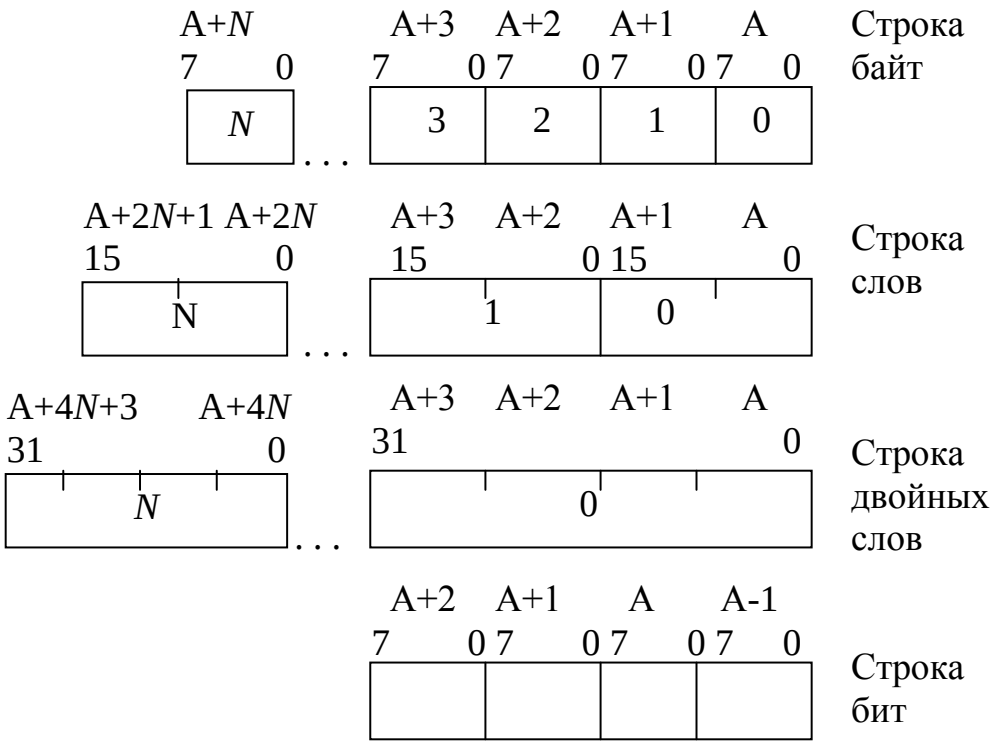


Рис. 2.5. Данные типа строка

Символьные данные

Поддерживаются строки символов в коде ASCII и арифметические операции (сложение, умножение) над ними (рис. 2.6). Поддержка осуществляется блоком ALU.

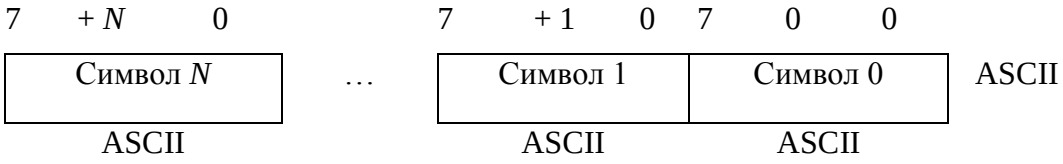


Рис. 2.6. Символьные данные

Данные типа указатель

Указатель содержит величину, которая определяет адрес фрагмента данных. Поддерживается два типа указателей, приведенных на рис. 2.7.

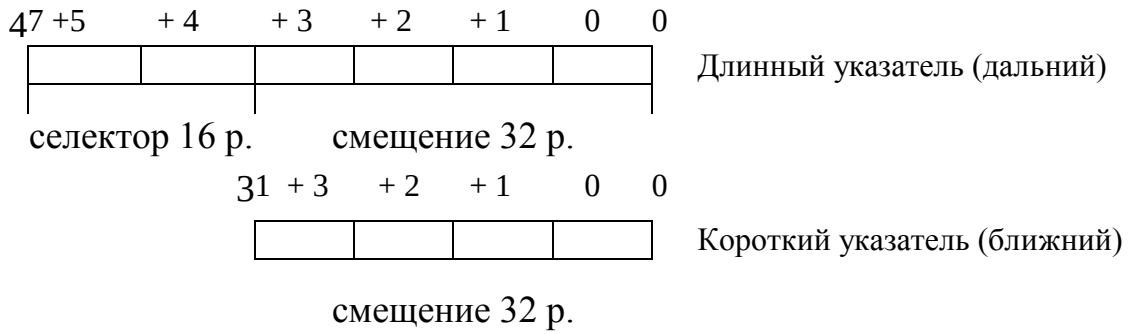


Рис. 2.7. Данные типа указатель

Векторные данные MMX-технологии

Целочисленные данные могут быть как со знаком, так и без знака (рис. 2.8).

Рис. 2.8. Данные MMX-технологии

Векторные данные SSE-расширения

На рис. 2.9 показаны 4 формата упакованных в 128 бит целочисленных данных, которые могут быть как со знаком, так и без знака.

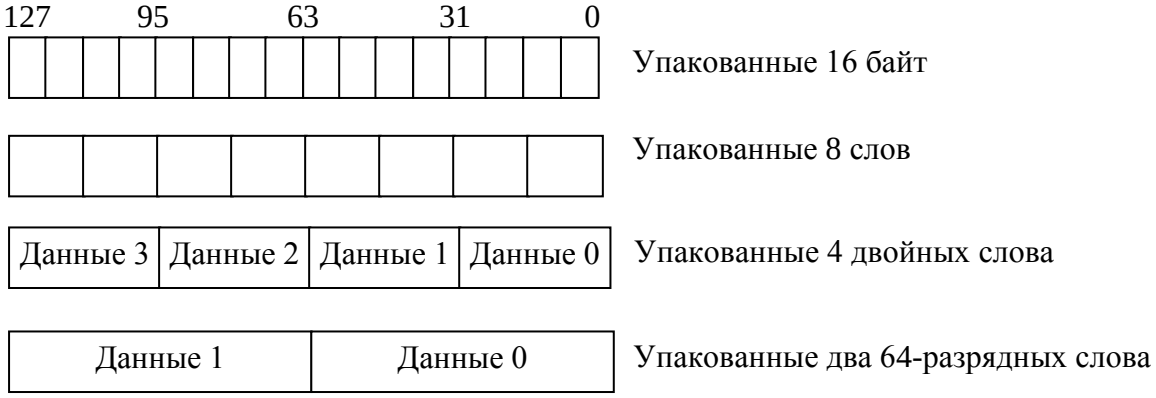


Рис. 2.9. Целочисленные данные SSE2 расширения

На рис. 2.10 приведен 128-разрядный формат упакованных данных с плавающей точкой одинарной и двойной точности.

Рис. 2.10. Данные с плавающей запятой: а – для SSE-расширения; а, б – для SSE2-расширения

Векторные данные AVX-расширния

На рис. 2.11. показаны два формата упакованных в 256 бит чисел с плавающей запятой одинарной и двойной точности.

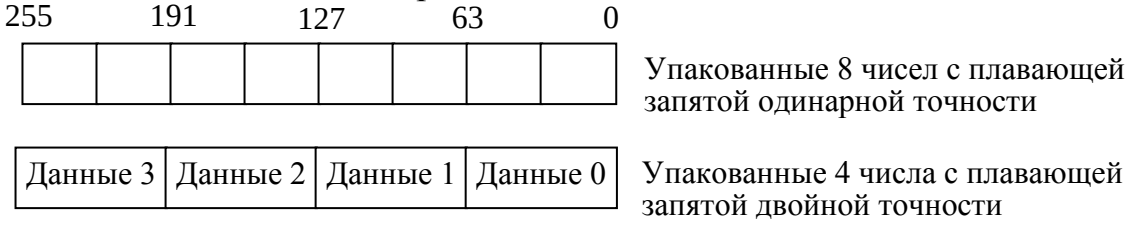


Рис. 2.11. Данные для AVX, AVX2-расширения

В AVX2-расширении кроме операций над числами с плавающей запятой, рассмотренных выше, выполняются логические операции OR, XOR, AND и т.д. над целочисленными данными (рис. 2.12). Арифметических операций нет.

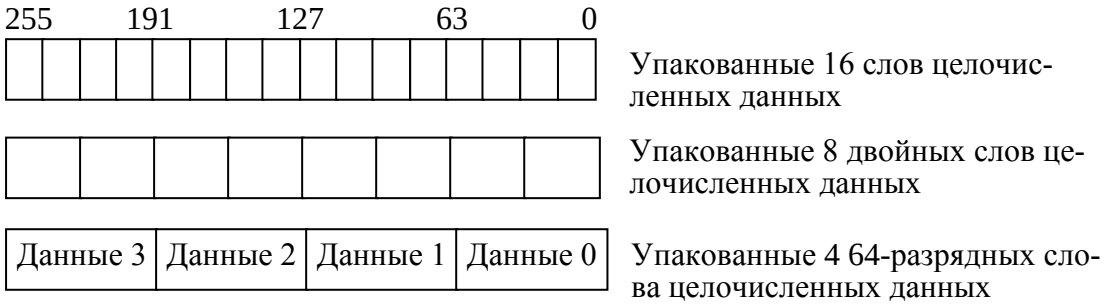


Рис. 2.12. Данные AVX2-расширения

Векторные данные MIC-расширения

На рисуке 2.13. показаны два формата упакованных в 512 бит целочисленных данных и два формата чисел с плавающей запятой одинарной и двойной точности.

Рис. 2.13. Данные MIC-расширения

Данные в IA-64

В IA-64 непосредственно поддерживается 6 типов данных, в том числе три формата, используемых ранее (одинарная точность, двойная

точность, расширенная точность), 82-разрядный формат данных с плавающей точкой (рис. 2.14) и 64-разрядные целые – со знаком и без знака.

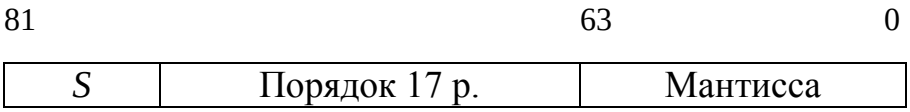


Рис. 2.14. Формат данных с плавающей точкой 82-разрядный

Теги и дескрипторы. Самоопределяемые данные

Одним из эффективных средств совершенствования архитектуры ЭВМ является теговая организация памяти, при которой каждое хранящееся в памяти (или регистре) слово снабжается указателем **– тегом** (рис. 2.15, *а*). Последний определяет тип данных (целое двоичное число, число с плавающей точкой, десятичное число, адрес, строка символов, дескриптор и т.д.), длину (формат) данных и некоторые другие их параметры. Теги формируются компилятором.

В интеловских процессорах теговая организация используется в кэш-памяти и блоках обработки чисел с плавающей запятой.

Дескрипторы – служебные слова, содержащие описание массивов данных и команд.

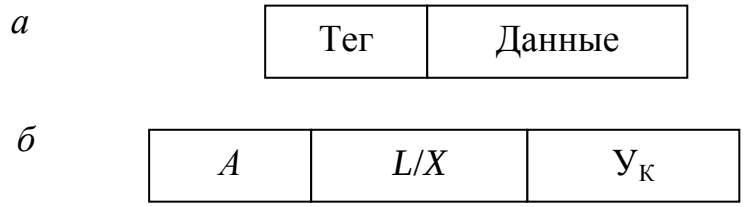


Рис. 2.15. Структура описания данных:
а – с теговой организацией памяти; *б* – дескриптор данных

Дескриптор содержит сведения о размере массива данных, его местоположении (в ОП или внешней памяти), адресе начала массива, типе данных, режиме защиты данных (например, запрет записи в ячейки массива) и некоторых других параметрах данных. Отметим, что задание в дескрипторе размера массива позволяет контролировать выход за границу массива при индексации его элементов. На рис. 2.15, *б* в качестве примера представлен один из видов дескрипторов – дескриптор данных.

Дескриптор содержит: *A* – адрес начала массива данных; *L* – длину массива; *X* – индекс; Ук – группу указателей (атрибутов).

Использование в архитектуре ЭВМ дескрипторов подразумевает, что обращение к информации в памяти производится через дескрипто-

ры, которые при этом можно рассматривать как дальнейшее развитие аппарата косвенной адресации.

Адресация информации в памяти может осуществляться с помощью цепочки дескрипторов, при этом реализуется многоступенчатая косвенная адресация. Более того, сложные многомерные массивы данных (таблицы и т.п.) эффективно описываются древовидными структурами дескрипторов.

2.3. Структура и форматы команд ЭВМ

Все возможные преобразования дискретной информации могут быть сведены к четырем основным видам:

- передача информации в пространстве (из одного блока ЭВМ в другой);
- передача информации во времени (хранение);
- логические (поразрядные) операции;
- арифметические операции.

Величины, над которыми выполняются операции, могут быть скалярными (принимающими в каждый момент времени только одно значение) и векторными.

ЭВМ, являющаяся универсальным преобразователем дискретной информации, выполняет указанные виды преобразований.

Обработка информации (решение задач) в ЭВМ осуществляется автоматически путем программного управления. Программа представляет собой алгоритм обработки информации (решение задачи), записанный в виде последовательности команд, которые должны быть выполнены машиной для получения результата.

Команда представляет собой код, определяющий операцию и данные, участвующие в операции.

По характеру выполняемых операций различают следующие основные группы команд:

а) команды арифметических операций над числами с фиксированной и плавающей точками;

б) команды десятичной арифметики;

в) команды логических операций и сдвигов;

г) команды передачи кодов;

д) команды операций ввода/вывода;

е) команды передачи управления;

ж) команды векторной обработки;

з) команды задания режима работы машины и др.

Команда в общем случае состоит из операционной и адресной частей (рис. 2.16, *а*).

В свою очередь, эти части, что особенно характерно для адресной части, могут состоять из нескольких полей.

Операционная часть содержит код операции (КОП), который задает вид операции (сложение, умножение и др.). Адресная часть содержит информацию об адресах операндов и результате операции.

Структура команды определяется составом, назначением и расположением полей в команде.

Форматом команды называют ее структуру с разметкой номеров разрядов (бит), определяющих границы отдельных полей команды, или с указанием числа бит в определенных полях.

Важной и сложной проблемой при проектировании ЭВМ является выбор структуры и форматов команды, т.е. ее длины, назначения и размерности отдельных ее полей. Естественно стремление разместить в команде в возможно более полной форме информацию о предписываемой командой операции. Однако в условиях, когда в современных ЭВМ значительно возросло число выполняемых различных операций и соответственно команд (в системе команд x86 более 500 команд) и значительно увеличилась емкость адресуемой основной памяти (4 Гбайт, 6 Гбайт), это приводит к недопустимо большой длине формата команды.

Действительно, число двоичных разрядов, отводимых под код операции, должно быть таким, чтобы можно было представить все выполняемые машинные операции. Если ЭВМ выполняет M различных операций, то число разрядов в коде операции

$$n_{\text{коп}} \geq \log_2 M; \text{ например, при } M = 500\ n_{\text{коп}} = 9.$$

Если основная память содержит S адресуемых ячеек (байт), то для явного представления только одного адреса необходимо в команде иметь адресное поле для одного операнда с числом разрядов

$$n_{\text{A}} \geq \log_2 S; \text{ например, при } S = 4 \text{ Гбайт } \ n_{\text{A}} = 32.$$

Отмечавшиеся ранее, характерные для процесса развития ЭВМ расширение системы (наборы) команд и увеличение емкости основной памяти, а особенно создание микроЭВМ с коротким словом, потребовали разработки методов сокращения длины команды. При решении этой проблемы существенно видоизменилась структура команды, получили развитие различные способы адресации информации.

Проследим изменения классических структур команд.

Чтобы команда содержала в явном виде всю необходимую информацию о задаваемой операции, она должна, как это показано на рис. 2.16, *б*, содержать следующую информацию:

A_1, A_2 – адреса операндов, A_3 – адрес результата, A_4 – адрес следующей команды (принудительная адресация команд).

Такая структура приводит к большой длине команды (например, при $M = 500$, $S = 4$ Гб длина команды – 137 бит) и неприемлема для прямой адресации операндов основной памяти. В компьютерах с RISC-архитектурой четырехадресные команды используются для адресации операндов, хранящихся в регистровой памяти процессора.

Можно установить, что после выполнения данной команды, расположенной по адресу K (и занимающей L ячеек), выполняется команда из $(K + L)$-й ячейки. Такой порядок выборки команды называется естественным. Он нарушается только специальными командами передачи управления. В таком случае отпадает необходимость указывать в команде в явном виде адрес следующей команды.

В трехадресной команде (рис. 2.16, *в*) первый и второй адреса указывают ячейки памяти, в которых расположены операнды, а третий определяет ячейку, в которую помещается результат операции.

Можно условиться, что результат операции всегда помещается на место одного из операндов, например первого. Получим двухадресную команду (рис. 2.16, *г*), т.е. для результата используется подразумеваемый адрес.

В одноадресной команде (рис. 2.16, *д*) подразумеваемые адреса имеют уже и результат операции, и один из операндов. Один из операндов указывается адресом в команде, в качестве второго используется содержимое регистра процессора, называемого в этом случае регистром результата, или аккумулятором. Результат операции записывается в тот же регистр.

Наконец, в некоторых случаях возможно использование безадресных команд (рис. 2.16, *е*), когда подразумеваются адреса обоих операндов и результата операции, например при работе со стековой памятью.

С точки зрения программиста, наиболее естественны и удобны трехадресные команды. Однако из-за необходимости иметь большее число разрядов для представления адресов основной памяти и кода операции длина трехадресной команды становится недопустимо большой и ее не удается разместить в машинном слове. Следует отметить, что очень часто в качестве операндов используются результаты предыдущих операций, хранимые в регистрах машины. По указанным причинам в современных ЭВМ применяют трехадресные команды для адресации

регистров. Обычно в ЭВМ используется несколько структур и форматов команд.

Приведенные на рис. 2.16 структуры команд достаточно схематичны. В действительности адресные поля команд большей частью содержат не сами адреса, а только информацию, позволяющую определить действительные (исполнительные) адреса операндов в соответствии с используемыми в командах способами адресации.

Рис. 2.16. Структуры команд:
а – обобщенная; *б* – четырехадресная; *в* – трехадресная;
г – двухадресная; *д* – одноадресная; *е* – безадресная

2.4. Способы адресации информации в ЭВМ

Существует два различных принципа поиска операндов в памяти: **ассоциативный** и **адресный**.

Ассоциативный поиск операнда (рис. 2.17) предполагает одновременный просмотр содержимого всех ячеек памяти для выявления кода, содержащего заданной командой ассоциативный признак (тег). Этот код и выбирается из памяти в качестве искомого операнда. В современных компьютерах ассоциативная выборка используется в кэш-памяти.

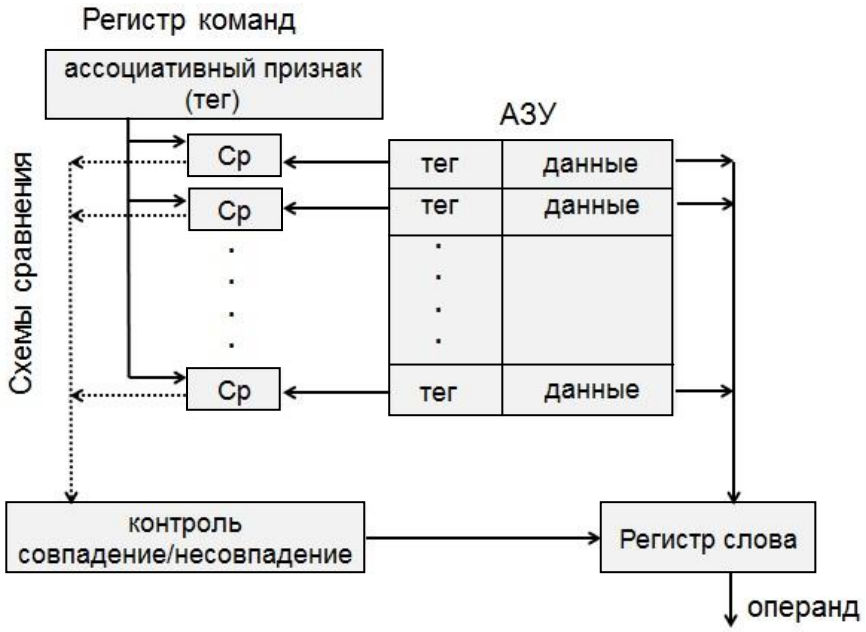


Рис. 2.17. Ассоциативный поиск

Адресный поиск предполагает, что искомый операнд извлекается из ячейки, номер которой формируется на основе информации в адресном поле команды (рис.2.18).

Рис. 2.18. Адресный поиск

Ниже мы будем рассматривать только реализацию адресного принципа поиска операнда. Следует различать понятия «адресный код» в ко-

манде $А_К$ и «исполнительный (физический) адрес» $А_И$. **Адресный код** – это информация об адресе операнда, содержащаяся в команде. **Исполнительный адрес** – это номер ячейки памяти, к которой производится фактическое обращение. В современных ЭВМ адресный код, как правило, не совпадает с исполнительным адресом. Таким образом, способ адресации можно определить как способ формирования исполнительного адреса операнда $А_И$ по адресному коду команды $А_К$.

Способов адресации существует много. Параметры процесса обработки информации существенно зависят от выбранного способа адресации. Одни способы адресации позволяют увеличить объём адресуемой памяти без удлинения команды, но снижают скорость выполнения операции, другие ускоряют операции над массивами данных, третьи – упрощают работу с подпрограммами и т.д.

В системах команд современных ЭВМ часто предусматривается возможность использования нескольких способов адресации операндов для одной и той же операции. Для указания способа адресации вводятся дополнительные разряды в поле кода операции, длина которого при этом возрастает.

Адресация операнда в команде может быть **явной** или **неявной.** При явной адресации в команде есть поле адреса операнда, в котором задается адресный код $А_К$. Большинство методов адресации являются явными.

При неявной адресации адресное поле в команде отсутствует, адрес операнда подразумевается кодом операции. Метод неявной адресации операндов используется во всех процессорах. Основное его назначение – уменьшение длины команды за счет исключения части адресов. При этом методе код операции точно задает адрес операнда. Например, из команды исключается адрес приемника результата. При этом подразумевается, что результат в этой команде помещается на место второго операнда.

Способы формирования адресов ячеек памяти ($А_И$) можно разделить на абсолютные и относительные.

2.4.1. Абсолютные способы формирования исполнительного адреса

Абсолютные способы формирования предполагают, что двоичный код адреса ячейки памяти ($А_И$) может быть извлечен целиком из адресного поля команды или из какой-либо другой ячейки (регистра), никаких преобразований над кодом адреса не производится.

К абсолютным способам относятся непосредственная, прямая и косвенная адресации, которые имеют различную кратность обращения (R) к памяти.

Непосредственная адресация операнда

При этом способе операнд располагается в адресном поле команды. Обращение к регистровой памяти (РП) или ОП за операндом не производится (R = 0), он выбирается вместе с командой. Таким образом, уменьшается время выполнения операции, сокращается объем памяти. Непосредственная адресация удобна для задания констант, длина которых меньше или равна длине адресного поля команды.

Прямая адресация операндов

При этом способе (рис. 2.19) адресации обращение за операндом в РП или ОП производится по адресному коду в поле команды (R = 1), т.е. исполнительный адрес операнда совпадает с адресным кодом команды ($A_И = A_К$).

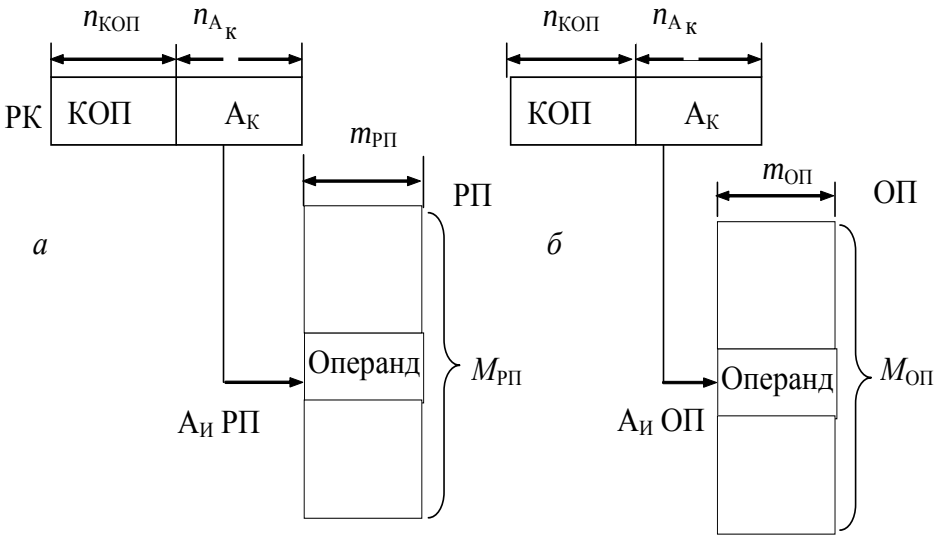


Рис. 2.19. Схема прямой адресации:
а – к регистровой памяти; *б* – к основной памяти

Обеспечивая простоту программирования, этот метод имеет существенный недостаток. Для адресации к ячейкам памяти большой емкости требуется «длинное» адресное поле в команде. Прямая адресация широко используется в сочетании с другими способами адресации. В частности, вся адресация к «малой» регистровой памяти ведется только с помощью прямой адресации.

Косвенная адресация операндов

При этом способе адресный код команды указывает адрес ячейки (регистра) памяти, в которой находится не сам операнд, а лишь адрес операнда, называемый указателем операнда. Адресация к операнду через цепочку указателей (косвенных адресов) называется косвенной ($R \geq 2$).

Адрес указателя, задаваемый программой, остается неизменным, а косвенный адрес может изменяться в процессе выполнения программы. Косвенная адресация, таким образом, обеспечивает переадресацию данных, т.е. упрощает обработку массивов и списковых структур данных, упрощает передачу параметров подпрограммам, но не обеспечивает перемещаемость программ в памяти.

Косвенная адресация также широко используется в ЭВМ, имеющих короткое машинное слово, для преодоления ограничений короткого формата. В этом случае первый указатель должен располагаться в РП (рис. 2.20).

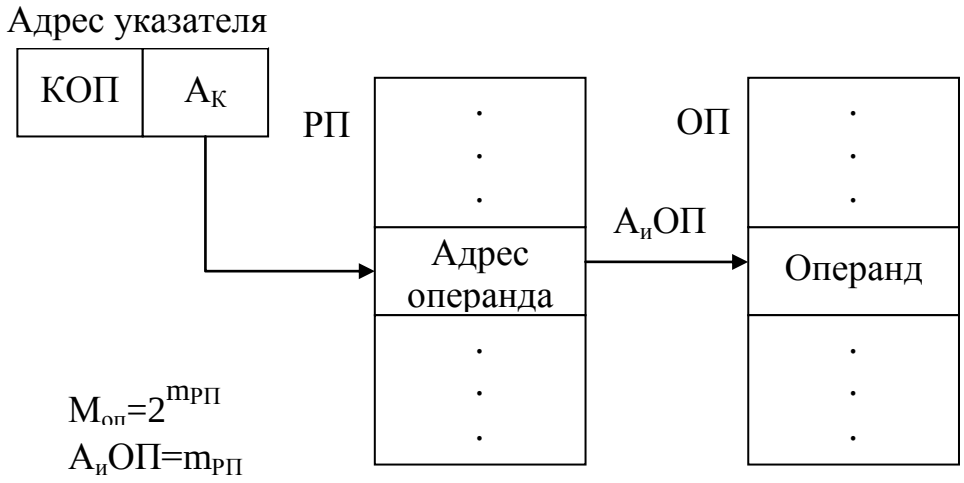


Рис. 2.20. Схема косвенной адресации

2.4.2. Относительные способы формирования исполнительных адресов ячеек памяти

Относительные способы формирования $A_И$ предполагают, что двоичный код адреса ячейки памяти образуется из нескольких составляющих: Б – код базы, И – код индекса, С – код смещения, используемых в сочетаниях (Б и С), (И и С), (Б, И и С).

При относительной адресации применяются два способа вычисления адреса $A_И$:

- суммирование кодов составляющих адреса;
- совмещение (конкатенация) кодов составляющих адреса.

Суммирование кодов составляющих производится для случаев:

$$А_И = Б + С;\ А_И = И + С;\qquad А_И = Б + И + С.$$

Базирование способом суммирования

В команде адресный код $А_К$ разделяется на две составляющие: $А_Б$ – адрес регистра регистровой памяти, в котором хранится база Б (базовый адрес); С – код смещения относительно базового адреса (рис. 2.21).

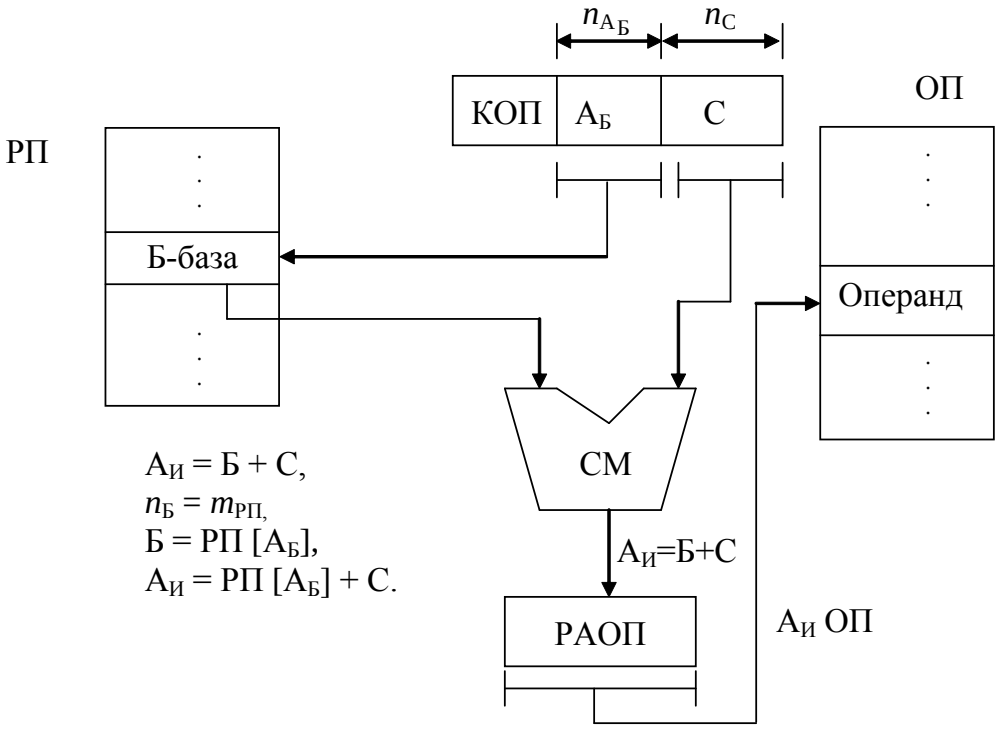


Рис. 2.21. Схема формирования относительного адреса способом суммирования кодов базы и смещения: СМ – сумматор; РАОП – регистр адреса ОП; Б – база (базовый адрес); С – смещение; $А_Б$ – адрес регистра базы; $n_Б$ – длина кода базы; $n_С$ – длина поля смещения

Для определения максимальной емкости ОП, адресуемой с помощью базирования способом суммирования, определим длину кода исполнительного адреса

$$n_{А_И} = n_{А_И ОП} = \max\{n_Б; n_С\}.$$

Так как $n_Б = m_{РП}$ и обычно больше, чем $n_С$, то справедливо следующее выражение:

$$M_{\text{ОП}} = 2^{n_{\text{Б}}} = 2^{m_{\text{РП}}},$$

т.е. максимальная адресуемая емкость ОП определяется разрядностью РП. Длина $n_{\text{А}_\text{Б}}$ поля кода команды, задающего адрес регистра базы $\text{А}_\text{Б}$, определяется через емкость РП $M_{\text{РП}}$ по формуле

$$n_{\text{А}_\text{Б}} \geq \log M_{\text{РП}}.$$

Таким образом, можно определить количество $n_{\text{А}_\text{К}}$ двоичных разрядов в адресном поле команды, необходимое для формирования $\text{А}_\text{И}$ с размещением базы в РП:

$$n_{\text{А}_\text{К}} = n_{\text{А}_\text{Б}} + n_{\text{С}} = \log_2 M_{\text{РП}} + n_C.$$

Приведенные выражения позволяют определить числовые значения параметров относительной адресации (базирование способом суммирования).

С помощью метода относительной адресации удается получить так называемый перемещаемый программный модуль, который одинаково выполняется процессором, независимо от адресов, в которых он расположен. При входе в модуль начальный адрес программного модуля (база) загружается в базовый регистр. Все остальные адреса программного модуля формируются через смещение относительно начального адреса (базы) модуля. Таким образом, одна и та же программа может работать с данными, расположенными в любой области памяти, без перемещения данных и без изменения текста программы, только за счет изменения содержания всего одного базового регистра.

Относительная адресация с совмещением составляющих $\text{А}_\text{И}$

Для увеличения емкости адресной ОП ($M_{\text{ОП}}$) без увеличения длины адресного поля команды $n_{\text{А}_\text{К}}$ можно использовать для формирования исполнительного адреса совмещение (конкатенацию) кодов базы и смещения (рис. 2.22).

При совмещении кодов базы и смещения

$$n_{\text{А}_\text{И}} = n_{\text{Б}} + n_{\text{С}}.$$

Таким образом, $M_{\text{ОП}} = 2^{n_{\text{А}_\text{И}}} = 2^{n_{\text{Б}}+n_{\text{С}}}$.

Следует отметить, что адресное пространство ОП может быть увеличено в $2^{n_{\text{С}}}$ раз за счет использования способа совмещения. Однако в данном случае начальные адреса массивов не могут быть реализованы произвольно, а должны иметь в младших разрядах $n_{\text{С}}$ нулей.

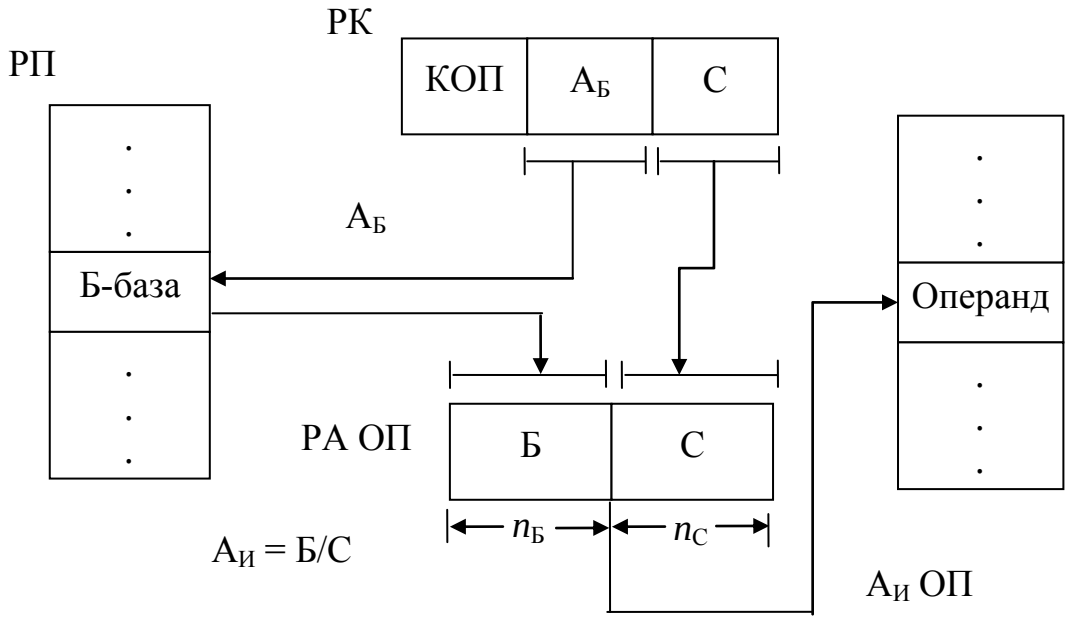


Рис. 2.22. Схема формирования относительного адреса способом совмещения кодов базы и смещения

Индексная адресация

Для работы программ с массивами, требующими однотипных операций над элементами массива, удобно использовать индексную адресацию. Схема индексной адресации аналогична базированию путем суммирования (рис. 2.23). В этом случае адрес *i*-го операнда в массиве определяется как сумма начального адреса массива (задаваемого полем смещения С) и индекса И, записанного в одном из регистров РП, называемом теперь индексным регистром. Адрес индексного регистра задается в команде полем адреса индекса – $A_{ИН}$ (аналогично $A_Б$).

В каждом *i*-м цикле содержимое индексного регистра изменяется на величину постоянную (часто равную 1). Использование индексной адресации значительно упрощает программирование циклических алгоритмов.

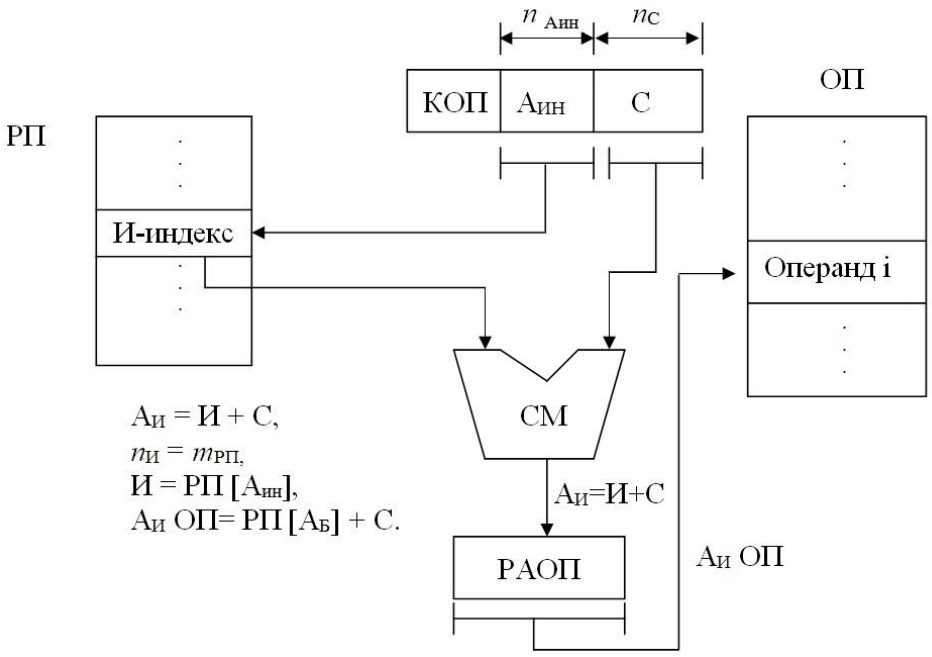


Рис. 2.23. Схема индексной адресации

Базово-индексная адресация со смещением

Для эффективной работы при относительной адресации применяется комбинированная индексация с базированием, при которой адрес операнда вычисляется как сумма трех величин (рис. 2.24):

$$A_И ОП = Б + И + С.$$

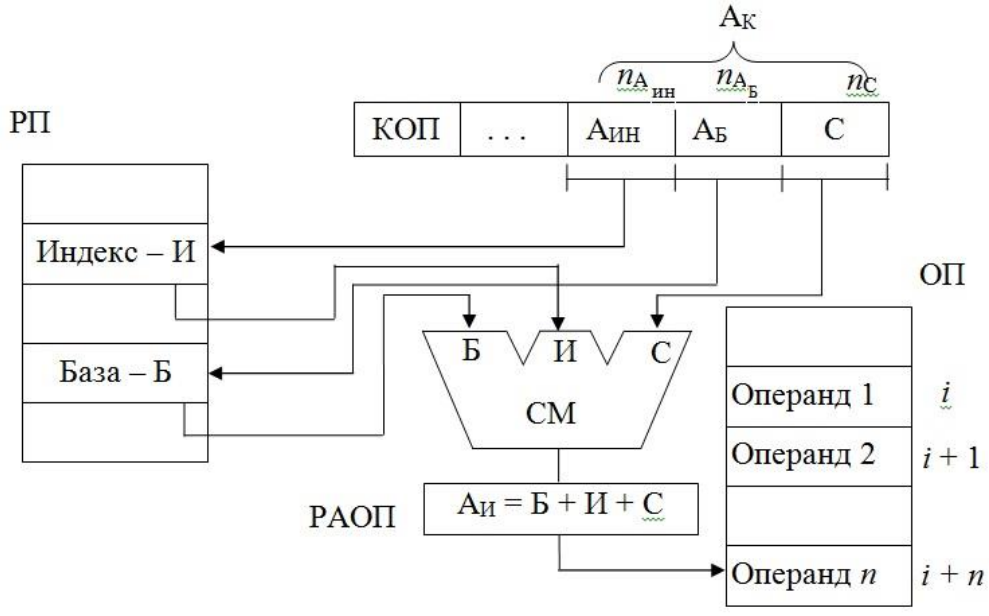


Рис. 2.24. Схема формирования исполнительного адреса
при базово-индексной адресации со смещением

Стековая адресация

Стековая память (стек) является эффективным элементом современных ЭВМ, реализует неявное задание адреса операнда. Хотя адрес обращения в стек отсутствует в команде, он формируется схемой управления автоматически по специальному правилу.

2.5. Примеры форматов команд и способов адресации

2.5.1. Форматы команд и способы адресации в интеловских процессорах

В качестве примера рассмотрим набор команд и способы адресации, используемые в процессорах интеловской архитектуры. Для этих процессоров в табл. 2.1 приведены данные о развитии их системы команд.

Таблица 2.1

Развитие системы команд x86 процессоров

Год появления набора команд	Тип процессора, где набор был реализован впервые	Общее число команд	Смысл расширения
1979	i8086	170	Исходный набор команд x86
1985	i386	220	50 новых команд для перехода к архитектуре IA-32
1997	Pentium/MMX	277	57 MMX-команд
1999	Pentium III	347	70 команд SSE-расширения, AMD 3DNow!
2000	Pentium 4 Northwood	491	144 команды SSE2
2004	Pentium 4 Prescott	504	13 команд SSE3, AMD x86-64
		514	10 команд Intel VT-x, AMD-V, Intel EM64T
2006	Core2 Duo (65 нм)	546	32 команды SSSE3
2007	Penryn (45 нм)	593	47 команд SSE4.1
2009	Core i7 Nehalem (45нм)	600	7 команд SSE4.2
2010	Core i5 Westmere (32 нм)	606	6 команд AES-NI
2011	Core i7, i5 Sandy Bridge (32 нм)		AVX
2013	Core i7, i5 Haswell (22 нм)		AVX2, FMA3, BMI
2015	Core i7, i5 Skylake (14 нм)		MIC, SGX

Изначально в базовом наборе команд процессора i8086 были предусмотрены команды обработки чисел с плавающей запятой, кото-

рые до i386-процессора включительно выполнялись на дополнительном сопроцессоре. Начиная с процессора i486, блок обработки чисел с плавающей запятой (FPU) стал составной частью микропроцессора.

Переход на 32-разрядную интеловскую архитектуру (IA-32) был осуществлен в процессоре i386 с добавлением 50 новых команд. Все последующие модели процессоров (до Pentium 4 включительно) имеют IA-32 архитектуру, несмотря на то, что расширение системы команд происходило неоднократно.

Технология виртуализации Intel VT-x, разработанная для IA-32 и поддерживаемая современными процессорами, использует 10 новых инструкций VM.

В 2002 г. впервые со времен i386 архитектура x86 подвергается принципиальным изменениям. Разработчиками фирмы AMD была создана 64-разрядная архитектура, получившая название «x86-64». Эта архитектура базируется на существующей архитектуре IA-32. В 2004 г. Intel вводит в серверные процессоры Xeon 64-разрядную технологию EM64T (Extended Memory), с программной точки зрения практически идентичную той, что предложила AMD. С 2006 г. эта технология под названием «Архитектура Intel 64» начала использоваться в клиентских ПК с процессором Core 2 Duo.

Начиная с 1997 г. и по сей день система команд x86 расширяется за счет SIMD инструкций: MMX, SSE, SSE2, SSE3, SSSE3, SSE4, AVX, AVX2, MIC.

В 2010 г. в процессорах Intel семейства Westmere (32 нм) стали использоваться 6 новых инструкций SIMD, которые Intel назвал AES-NI.

MMX-технология

В основе технологии MMX лежит расширение набора команд (57 новых команд) для эффективного выполнения типичных мультимедийных алгоритмов, к числу которых относятся и многие алгоритмы, характерные для цифровой обработки сигналов. Это первое существенное изменение в системе команд микропроцессоров семейства x86, начиная с выхода в свет микропроцессора i386. В технологии MMX использована модель обработки данных SIMD, предусматривающая одновременное выполнение операции над несколькими целочисленными операндами разрядностью 1, 2 или 4 байта.

MMX-команды используют восемь 64-разрядных MMX-регистров с плавающей запятой и реализуются в том же режиме процессора, что и команды с плавающей запятой. MMX-команды делятся на следующие группы: арифметические, логические и сдвига, сравнения, передачи

данных, упаковки и распаковки, отмены режима MMX. Все программное обеспечение, созданное для ранее выпущенных процессоров, без всяких изменений может выполняться на процессорах с технологией MMX.

Стремясь устранить недостатки, свойственные MMX-технологии (отсутствие MMX-команд для работы с плавающей запятой, невозможность выполнения операций с плавающей запятой при выполнении MMX-команд), Intel решила внести необходимые дополнения в архитектуру процессора Pentium III.

SSE-расширение

Новые 70 команд SSE-расширения делятся на 4 категории:

- SIMD-команды обработки данных в формате с плавающей запятой одинарной точности (SPFP-команды);
- дополнительные SIMD-команды для обработки целочисленных данных;
- команды управления кэшированием;
- команды сохранения и восстановления состояния процессора.

SPFP-команды используют 8 новых 128-разрядных регистров (XMM-регистры) и новый тип данных – 128-разрядное значение, содержащее 4 последовательно расположенных («упакованных») 32-разрядных числа с плавающей запятой одинарной точности. При выполнении инструкций с XMM традиционное оборудование FPU/MMX не используется, что позволяет эффективно смешивать инструкции MMX с инструкциями над операндами с плавающей точкой.

Большинство SPFP-команд имеют два операнда. Данные, содержащиеся в первом операнде, после выполнения команды, как правило, замещаются результатами, а данные, содержащиеся во втором операнде, остаются неизменными.

SPFP-команды поддерживают два типа операций над упакованными данными с плавающей запятой – параллельные и скалярные. Параллельные операции выполняются над четырьмя 32-разрядными элементами данных, упакованными в каждый 128-разрядный операнд.

Скалярные операции выполняются над младшими (занимающие разряды 0–31) элементами данных двух операндов. Остальные три элемента данных не изменяются.

В расширение SSE включены дополнительные SIMD-команды для работы с целочисленными данными. Эти новые команды расширяют возможности существующего набора команд технологии MMX. Они выполняют SIMD-операции над несколькими целочисленными данны-

ми, упакованными в 64-разрядные группы, загружают и хранят упакованные данные в 64-разрядных MMX-регистрах.

Кроме того, в SSE введены команды нового типа, обеспечивающие:

- управление кэшированием данных с целью повышения эффективности использования кэш-памяти и сокращения числа обращений к основной памяти;
- упреждающее кэширование данных с целью организации параллельной работы конвейера команд и обмена с памятью.

Первая группа команд выполняет запись данных из MMX (XMM) регистра в память, минуя кэш. Вторая – обеспечивает запись данных из памяти в кэш различных уровней.

Кроме XMM-регистров в микропроцессоре Pentium III появился новый регистр состояния и управления MXCSR. Для работы с этими регистрами требуется поддержка как со стороны процессора, так и со стороны операционной системы. Чтобы прикладные программы и ОС могли сохранять и восстанавливать состояния новых компонентов процессора, введено несколько команд управления. Первая группа команд управления обеспечивает сохранение в памяти содержимого регистра MXCSR и, наоборот, загружает слово состояния из памяти в регистр MXCSR. Вторая группа – сохраняет в памяти состояние процессора (состояние регистров данных FPU, MMX-регистров, XMM-регистров) и восстанавливает ранее сохраненное состояние процессора.

Расширения SSE2, SSE3, SSSE3, SSE4

Расширение SSE2, введенное в состав Pentium 4 Northwood, значительно расширяет возможности обработки нескольких операндов по принципу SIMD по сравнению с SSE. В нем используется 144 новых команды, обеспечивающих одновременное выполнение операций над несколькими операндами, которые располагаются в памяти и в 128-разрядных регистрах XMM. В регистрах могут храниться и одновременно обрабатываться два числа с плавающей запятой в формате двойной точности (64 разряда) или 4 числа в формате одинарной точности (32 разряда), любые целочисленные типы данных, способные разместиться в 128-разрядных регистрах.

Расширение SSE2, представляя собой симбиоз MMX и SSE, обладает большей гибкостью и позволяет добиваться впечатляющего прироста производительности. Команды SSE2 существенно повышают эффективность процессора при реализации трехмерной графики и Интер-

нет-приложений, обеспечение сжатия и кодирования аудио- и видеоданных и в ряде других приложений.

Расширение SSE3, введенное в состав Pentium 4 Prescott, включает 5 новых операций с комплексными числами, 5 потоковых операций над числами с плавающей запятой, 2 команды для синхронизации потоков и одну специальную инструкцию для применения при кодировании видео.

Расширение SSSE3 (Supplemental SSE3 – дополнительное потоковое SIMD-расширение 3) появилось в процессорах с микроархитектурой Intel Core и поддерживается процессором Intel Atom. Новыми в SSSE3, по сравнению с SSE3, являются 16 уникальных команд, работающих с упакованными целыми данными. Каждый из них может работать как с 64-битными (MMX), так и с 128-битными (XMM) регистрами, поэтому Intel в своих материалах ссылается на 32 новые команды. Новые инструкции включают работу со знаком, сдвиги, перемешивание байт, умножение, горизонтальное сложение/вычитание целых.

Расширение SSE4.1 появилось в первом процессоре Intel с 45 нм техпроцессом (кодовое наименование Penryn). Набор команд SSE4.1 включает 47 новых инновационных инструкций, основными из которых являются примитивы векторизации для компиляторов и ускорители кодирования видеозаписей с высоким расширением и обработки фотоизображений.

Расширение SSE4.2 разработано Intel для процессоров с новой микроархитектурой Nehalem. Введенные в набор SSE4.2 инструкции ориентированы на ускорение обработки строк и текстовой информации.

Ни одна из SSE4 инструкций не работает с 64-битными MMX-регистрами, только с 128-битными XMM-регистрами.

Расширения AES-NI

Расширение AES-NI (Advanced Encryption Standard New Instructions) – набор из 6 новых SIMD-инструкций, ускоряющий процесс шифрования и дешифрования информации по стандарту AES. Стандарт AES является стандартом шифрования США, принятым в 2000 г. Он специфицирует алгоритм Rijndael, который представляет собой симметричный блочный шифр, работающий с блоками длиной 128 бит, и использует ключи длиной 128, 192 и 256 бит. По заявлению правительства США для взлома шифрования при использовании 128-битного ключа потребуется 149 триллионов лет.

Расширения AVX

Новое расширение AVX (Advanced Vector Extensions) – расширение системы команд x86 для микропроцессоров с новой микроархитектурой Intel Sandy Bridge (2010 г.) и процессоров AMD Bulldozer (2011 г.),

анонсированная Intel в 2008 г., представляет различные улучшения, новые инструкции и новую схему кодирования машинных кодов. Размер векторных регистров SIMD увеличивается со 128 (XMM) до 256 бит (регистры YMM). Существующие 128-битные инструкции будут использовать только младшую половину новых YMM-регистров. В будущем возможно расширение до 512 или 1024 бит. Набор инструкций AVX позволяет использовать любую двухоперандную инструкцию XMM в трехоперандном виде без модификации двух регистров-источников, с отдельным регистром для результата. Добавлены инструкции с количеством операндов более трех. Новая система кодирования машинных кодов VEX предоставляет новый набор префиксов кода, которые расширяют пространство возможных машинных кодов. Использование YMM-регистров поддерживают операционные системы: Windows 7, Windows Server 2008 R2, Linux (версия ядра 2.6.30).

Расширение AVX подходит для интенсивных вычислений с плавающей точкой в мультимедийных, научных и финансовых задачах. Увеличивает степень параллелизма и пропускную способность в вещественных SIMD-вычислениях.

Расширения AVX2, FMA3, BMI

Набор инструкций AVX2 является расширением набора AVX и используется в процессорах с новой микроархитетурой Haswell. Главное отличие нового набора инструкций AVX2 от прежней версии AVX заключается в том, что если ранее 256-битные операции с AVX регистрами были доступны только для операнда с плавающей запятой, а для целочисленных операндов были доступны лишь 128-битные операции, то в AVX2 256 операции стали доступны и для целочисленных операндов.

Кроме того в AVX2 появилась улучшенная поддержка сдвигов и перестановок в векторных операциях. Есть и новые инструкции, используемые для сборки нескольких (4-х или 8-ми) несвязанных элементов в один векторный элемент, благодаря чему есть возможность более полно загружать 256-битные AVX регистры.

Новый **набор инструкций FMA3 (Fused Multiply Add)** используется в процессорах с новой микроархитетурой Haswel и предназначен для проведения операций совмещённого умножения и сложения над 3-мя операндами. Использование операций FMA3 позволяет более эффективно реализовать операции деления, извлечение квадратного корня, умножения векторов и матриц и т.д. Набор FMA3 включает 36 инструк-

ций с плавающей запятой для выполнения 256-битных вычислений и 60 инструкций для 128-битных векторов.

В **набор команд BMI** (Bit Manipulation Instructions) входят 15 скалярных инструкций для битовых операций, которые работают с целочисленнами регистрами общего назначения.

Эти инструкции разбиты на три группы:

- манипуляции на отдельными битами, такие как вставка, сдвиг и извлечение бит;
- подсчет битов, например, подсчет ведущих нулей в записи чисел;
- целочисленное умножение произвольной точности.

Данный набор инструкций позволяет ускорять ряд специфических операций, используемых, например, при шифровании.

Расширения MIC, SGX

В 2015 г. прошла презентация новой микроархитектуры процессоров Intel Skylake с техпроцессом 14 нм. В презентации прозвучали достаточно любопытные откровения о том, что построенные на Intel Skylake серверные и клиентские процессоры могут серьезно различаться по своей конфигурации даже на уровне микроархитектуры. Один пример такого отличия уже хорошо известен – серверные Skylake получили поддержку расширения системы команд MIC (AVX-512), которое в остальных процессорах не используется.

Расширение SIMD инструкций MIC (Many Integrated Core Architecture) использует шестнадцать 512-битных регистров для выполнения векторных операций над целочисленными данными и данными с плавающей запятой одинароной и двойной точности.

Нововведения в системе команд не миновали и клиентские процессоры. Так, в них появились новые инструкции семейства Intel SGX (Software Guard Extension). Входящие в этот набор команды позволяют приложению создать для своего исполнения изолированную и защищённую среду в памяти, доступ к которой будет невозможен ни для каких иных процессов и устройств. Таким образом, приложение, оперирующее критически важной информацией, сможет защитить свой код и данные от каких-либо программных и аппаратных атак и вторжений, что может поднять безопасность платформы x86 на новый уровень. Intel отдельно подчеркивает, что благодаря SGX можно создавать и полностью защищенный программный код, который невозможно отслеживать с помощью аппаратных отладчиков ITP-класса.

Обобщенный формат команд x86

Базовый набор команд 32-разрядного интеловского процессора обеспечивает выполнение операций над операндами, которые находятся в регистре, памяти или непосредственно в команде. В набор входят безадресные, одно-, двух- и трехадресные команды. Процессор реализует следующие шесть типов двухадресных команд: регистр – регистр; память – регистр; непосредственный операнд – регистр; регистр – память; память – память; непосредственный операнд – память.

Операнды могут содержать 8, 16 или 32 разряда. Для реализации различных типов команд определены форматы, задающие порядок размещения информации о выполняемой операции и способах выбора операндов.

Обобщенный вид формата команды показан на рис. 2.20. Он допускает наличие следующих полей: кода операции (1 или 2 байта); байтов адресации (0, 1 или 2 байта); байтов смещения (0, 1, 2 или 4 байта); байтов непосредственных данных – операндов (0, 1, 2 или 4 байта).

Команды содержат от 1 до 12 байт. Проведенные оценки показывают, что в среднем длина команды составляет 4–5 байт.

Рассмотрим назначение основных полей кода команды (рис. 2.25). Код операции (КОП) определяет тип выполняемой операции, а также в некоторых командах в первом байте может содержаться бит W, задающий разрядность операндов:

W = 0 – операция с байтами;

W = 1 – операция со словами (16 или 32 разряда).

КОП	Байты адресации		Смещение	Операнд
	MOD R/M	SIB		
1 или 2 байта	0 или 1 байт	0 или 1 байт	0, 1, 2 или 4 байта	0, 1, 2 или 4 байта

Рис. 2.25. Общий формат команд

В ряде команд первый байт КОП содержит поля reg или sreg, определяющие адреса используемых регистров. Трехбитовое поле reg задает выбираемый регистр в соответствии с разрядностью обрабатываемых операндов. Поле sreg (двух или трехбитовое) определяет адрес сегментных регистров.

Байт адресации MOD R/M содержит три поля (рис. 2.26). Поля MOD и R/M задают адрес одного из операндов, который может храниться в регистре или ячейке памяти. Кодировка этих полей определяет выбираемый способ адресации.

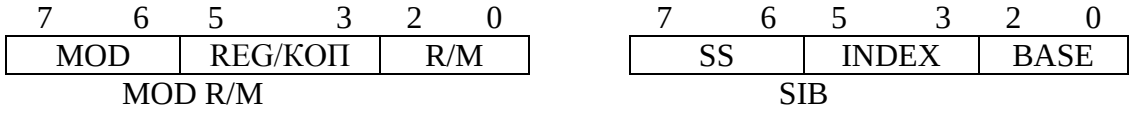


Рис. 2.26. Форматы байтов MOD R/M и SIB

В одноадресных командах поле REG/КОП содержит дополнительные биты кода операции. В двухадресных командах поле REG содержит адрес регистра, в котором хранится второй из операндов. Тип команды (одно- или двухадресная) определяется первым битом КОП. Поле MOD указывает, какой разрядности смещение используется для формирования адреса. Если оно имеет значение 00 (при некоторых значениях R/M) или 01, 10, то используется 8-, 16- или 32-разрядное смещение. Это смещение задается соответствующими байтами в коде команды, которые располагаются после байтов адресации.

Для реализации некоторых способов относительной адресации используется байт SIB. Он содержит 3-битовые поля INDEX и BASE, определяющие выбор регистров, используемых в качестве индексного и базового регистров, и поле SS, задающее масштабный коэффициент для модификации значения индекса.

При выполнении операций с непосредственной адресацией один из операндов задается в последних байтах команды (рис. 2.25). В этом случае КОП ряда команд содержит бит *S*, определяющий способ использования непосредственно задаваемых данных.

Способы адресации

Интеловский 32-разрядный процессор реализует сегментную организацию оперативной памяти, при которой физический адрес ячейки памяти формируется путем сложения базового адреса сегмента и относительного адреса ячейки внутри сегмента.

Базовый адрес определяется содержимым 16-разрядного сегментного регистра и зависит от режима работы процессора. Если он работает в режиме обработки 16-разрядных данных (режим реальных адресов), то 20-разрядный базовый адрес формируется путем сдвига содержимого сегментного регистра на 4 разряда влево. Если процессор работает в режиме обработки 32-разрядных данных (защищенный режим), то 32-разрядный базовый адрес содержится в дескрипторе, выбор которого из таблицы дескрипторов осуществляется с помощью селектора – содержимого соответствующего сегментного регистра.

В качестве относительного адреса используется содержимое регистров общего назначения или эффективный адрес (EA), который фор-

мируется в соответствии с заданным способом адресации. EA является 16- или 32-разрядным и формируется в зависимости от значения полей MOD и R/M и содержимого байта SIB (для 32-разрядных адресов). В общем случае EA образуется путем арифметического сложения трех компонентов:

- содержимого базового регистра;
- содержимого индексного регистра;
- 8-, 16-, 32-разрядного смещения, заданного в одном, двух или четырех байтах команды.

В зависимости от значений полей MOD и R/M для формирования EA используются все или часть этих слагаемых.

В процессоре осуществляются следующие способы адресации операндов:

- непосредственная адресация;
- регистровая адресация;
- косвенно-регистровая адресация;
- прямая адресация;
- базовая адресация;
- индексная адресация;
- базово-индексная адресация;
- базово-индексная адресация со смещением.

В современных микропроцессорах Intel Core i5, i7 базовый набор команд и используемые способы адресации операндов практически полностью совпадают с набором команд и способов адресации в предыдущих моделях – Core 2 Duo, Pentium 4. Процессоры обеспечивают реальный и защищенный режимы работы, реализуют сегментную и страничную организации памяти. Таким образом, пользователь имеет дело с хорошо знакомым набором регистров и способов адресации, может работать с базовой системой команд и известными вариантами реализации прерываний и исключений, которые характерны для всех моделей семейств Intel Core и Pentium.

Основные режимы работы x86-64 архитектуры

Как было сказано выше, корпорацией AMD было разработано 64-разрядное расширение x86-архитектуры, которое получило название «x86-64», т.е. 64-битная x86 (по аналогии с x86-32). Позднее x86-64 архитектура была переименована в AMD64. В отличие от 64-битной архитектуры IA-64, примененной в процессорах Intel Itanium, x86-64 базируется на существующей архитектуре x86-32. Следовательно, процес-

сор, построенный на основе x86-64, может безо всяких проблем исполнять существующие 32-битные приложения, которых написано на текущий момент просто немерено (и в них вложены очень большие деньги). Причем эти приложения могут выполняться без каких бы то ни было потерь в производительности в отличие от того же Intel Itanium, где x86-32 систему команд приходится эмулировать.

В процессорах данной архитектуры существующие в x86 регистры общего назначения расширены с 32 до 64 бит и к ним добавлены еще 8 новых 64-разрядных регистров.

Для реализации одновременной работы как с 32-битным, так и с 64-битным кодом и регистрами архитектура AMD64 предполагает поддержку процессорами двух режимов: **Long Mode («длинный» режим),** имеющего два подрежима – **64-битный режим** и **Compatibility mode (режим совместимости)**, и **Legacy Mode (наследственный режим).** Что они собой представляют, можно понять из табл. 2.2.

Таблица 2.2

Режимы работы процессора x86-64 архитектуры

Режим		ОС	Необх. перекомпиляция приложений	Характеристики			
				Длина адреса	Длина операнда	Дополнительные регистры	Размер РОН
Long Mode	64-битный	64-битная	ДА	64	64	ДА	64
	Compatibility mode		НЕТ	32	32	НЕТ	32
Legacy Mode		32-бит.	НЕТ	32	32	НЕТ	32
		16-бит.		16	16		

Итак, в **64-битном режиме** обеспечивается поддержка:

- 64-битных виртуальных адресов;
- 8-ми новых и расширенных 64-битных регистров общего назначения;
- 64-битного указателя инструкций RIP;
- сплошного адресного пространства с единым пространством для инструкций, данных и стека.
- 64-битных арифметических и логических операций над целыми числами.

Данный режим снимает ограничение в размерности адресного пространства оперативной памяти, которое в современных 32-разрядных x86 системах составляет 2^{32} = 4 Гбайт.

Для адресации новых регистров в команды введены так называемые «префиксы расширения регистра», кодирование которых осу-

ществляется кодами, использующимися для команд INC <регистр> и DEC <регистр> в 32- и 16-битных режимах. Команды INC и DEC в 64-битном режиме должны кодироваться в более общей, двухбайтовой форме.

Compatibility mode обеспечивает бинарную совместимость с существующими 16- и 32-битными приложениями при работе с 64-битной операционной системой. Этот режим разрешается ОС по принципу отдельных кодовых сегментов. Однако, в отличие от 64-битного режима, сегментация функционирует обычным образом, используя семантику защищенного режима. С точки зрения выполняемого приложения процессор выглядит как обычный x86 центральный процессор (CPU) в защищенном режиме. С точки зрения операционной системы трансляция адресов, работа с прерываниями и исключениями, а также системные структуры данных используют механизмы 64-битного Long Mode.

Наследственный режим (Legacy Mode) обеспечивает бинарную совместимость с 16- и 32-битными операционными системами; полную совместимость с существующими 32-битными реализациями x86 архитектуры, включающей в себя поддержку сегментированной памяти и 32-битных регистров общего назначения и указателя инструкций; процессор уподобляется обычному 32-разрядному x86 CPU. В этом режиме не задействуется ни одна из 64-битных функций.

Нельзя не отметить, что для того, чтобы пользователи смогли воспользоваться преимуществами 64-битного режима, необходим компилятор, который разработан и поставляется вместе с микропроцессором.

Данная архитектура была реализована в процессорах AMD Athlon 64, Opteron.

Особенности архитектуры Intel 64

Архитектура Intel 64 (технология EM64T) в сочетании с соответствующим программным обеспечением поддерживает работу 64-разрядных приложений на серверах, рабочих станциях, настольных ПК и ноутбуках. Она, как и x86-64 от AMD, реализует 64-разрядное расширение регистров, те же режимы работы процессора, ту же программную совместимость с 16- и 32-битными приложениями, а главное – эта технология расширяет адресное пространство виртуальной и физической памяти.

Архитектура Intel 64 поддерживает следующие возможности:

- 64-разрядное сплошное пространство виртуальных адресов;
- 64-разрядные указатели;
- 64-разрядные регистры общего назначения;

- 64-разрядную поддержку вычислений с целыми числами;
- до 1 Тбайт адресного пространства платформы.

2.5.2. Форматы команд и способы адресации в RISC-процессорах

Рассмотрим форматы команд на примере процессоров архитектуры Power PC, разработанной корпорациями IBM, Apple и Motorola.

Все команды имеют длину 32 разряда и могут быть трех форматов:

- 1-й формат – КОП (6); R_S (5); R_t (5); I (16), где КОП(6) – поле кода операции, содержащее 6 разрядов; R_S(5), R_t(5) – поля адресов регистров (по 5 разрядов); I(16) – 16-разрядный непосредственный операнд;
- 2-й формат – КОП (6); R_S (5); R_t (5); R_k (5);
- 3-й формат отличается от 2-го формата наличием дополнительного 32-разрядного командного слова, в котором для различных кодов операций могут находиться 32-разрядные непосредственный операнд, смещение или адрес перехода.

Архитектура Power PC определяет операции типа *регистр – регистр* для всех команд обработки. Источником данных являются встроенные регистры или непосредственные операнды. Трехрегистровый формат команд позволяет отличать регистр результатов от двух регистров – источников, позволяя использовать их в других командах. Данные пересылаются между памятью и регистрами только специальными командами загрузки/сохранения. Адреса памяти формируются с использованием базового регистра и смещения.

2.5.3. Особенности системы команд IA-64

Шестидесятичетырехразрядная интеловская архитектура (IA-64), как было сказано выше, реализует EPIC-концепцию, разработанную совместно фирмами Intel и HP; IA-64 не является 64-разрядным расширением 32-разрядной архитектуры x86 компании Intel или переработкой 64-разрядной архитектуры PA-RISC компании HP; IA-64 представляет собой нечто абсолютно новое – передовую архитектуру, использующую длинные слова команд, предикаты команд, устранение ветвлений, предварительную загрузку данных и другие ухищрения для того, чтобы «извлечь больше параллелизма» из кода программ.

Команды IA-64 можно подразделить на команды работы со стеком регистров (например, **alloc**); целочисленные команды; команды сравнения и работы с предикатами; команды доступа в память; команды перехода; мультимедийные команды; команды пересылок между регистра-

ми; команды выполнения операций над строками и подсчет числа единиц в слове; команды работы с плавающей запятой.

Целочисленные команды IA-64 включают арифметические операции (**add**, **sub** и др.), операции над битами и сдвиги, а также 32-разрядные операции.

Отметим, что команда умножения целых чисел в регистрах общего назначения (GR) отсутствует; для перемножения необходима пересылка целых в регистры с плавающей запятой (FR) и применение операции умножения, выполняемой в функциональном исполнительном устройстве вещественного типа. Некоторые специалисты считают это «наименее удачной» чертой системы команд IA-64.

Команды сравнения и работы с предикатами – это одна из принципиально новых особенностей IA-64 по сравнению с RISC-архитектурой. Приведем несколько типичных примеров команд этой группы. Команда **cmp** сравнивает два регистра GR (или регистр GR и литерал) на одно из 10 возможных условий (больше, меньше или равно и т.п.). Команда **tbit** тестирует заданный бит GR. Команда **fcmp** сравнивает два числа с плавающей запятой. Однако результатом сравнения является не единственный код условия, что типично для обычных процессоров. Логический результат сравнения (1 – истина, 0 – ложь) записывается обычно в пару предикатных регистров (во второй пишется отрицание первого). Эти значения предикатных регистров (PR) используются затем не только в командах условного перехода, как в обычных микропроцессорах. Почти все команды IA-64 выполнимы «под предикатом», т.е. могут выполняться или нет в зависимости от значения указанного в команде PR-регистра. Это позволяет во многих случаях избежать применения условных переходов, которые, как известно, отрицательно сказываются на производительности процессоров. Вместо этого процессор с архитектурой IA-64, имеющий большое число ресурсов (в частности, регистров и функциональных исполнительных устройств), может исполнять обе ветви программы линейно.

Формат команд IA-64 содержит 41 разряд и имеет фиксированную длину (рис. 2.27). Поле КОП занимает 14 разрядов, под адрес 64 предикатных регистров (PR) отводится 6 разрядов, три 7-битных поля (GFR) используются для адресации 128 регистров общего назначения (GR) или регистров с плавающей точкой (FR).

Большинство целочисленных команд трехадресные, а их аргументы находятся в регистрах, однако встречается и литеральное (символьное) представление аргументов. Имеются также модификации команд **add**

и **sub**, которые являются четырехадресными: в них к сумме/разности регистров прибавляется/вычитается 1.

Рис. 2.27. Формат инструкций IA-64

Команды в формате IA-64 упакованы по три в 128-битный LIW (long instruction word) – пакет (рис. 2.28).

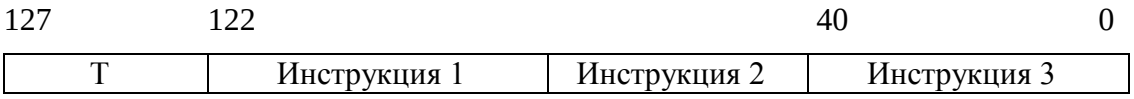


Рис. 2.28. Пакет инструкций IA-64

В каждый пакет при трансляции компилятор помещает шаблон, который размещается в 5-битовом поле T. Шаблон пакета указывает не только на то, какие команды в пакете могут выполняться независимо, но и какие команды из следующего пакета могут выполняться параллельно. Команды в пакетах не обязательно должны быть расположены в том же порядке, что и в машинном коде, и могут принадлежать к различным путям ветвления. Компилятор может также помещать в один пакет зависимые и независимые команды, поскольку возможность параллельного выполнения определяется шаблоном пакета. В отличие от некоторых ранее существовавших архитектур со сверх длинными словами команд (VLIW) IA-64 не добавляет команд «нет операции» (NOPS) для дополнения пакетов.

2.6. Принципы организации системы прерывания программ

Во время выполнения ЭВМ текущей программы внутри машины и в связанной с ней внешней среде (технологический процесс, управляемый ЭВМ) могут возникать события, требующие немедленной реакции на них со стороны машины.

Реакция состоит в том, что при поступлении сигнала на прерывание выполнение текущей последовательности команд приостанавливается и управление передается обработчику прерываний, который реагирует на события и обслуживает его, после чего возвращает управление в прерванный код программы.

Рассматриваемый процесс, называемый прерыванием программ, поясняется на рис. 2.29.

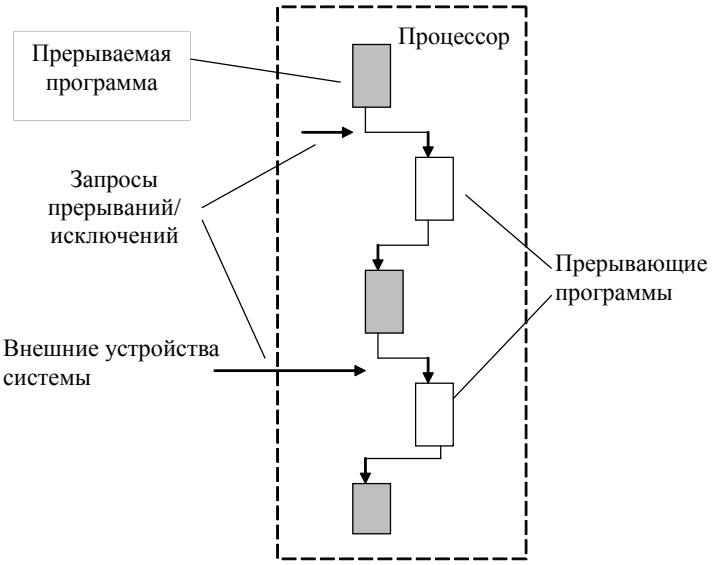


Рис. 2.29. Процесс прерывания программы

Обработчик прерываний – это программа обработки прерываний, являющаяся частью ОС, предназначенная для выполнения ответных действий на условие вызвавшее прерывание.

Типы прерываний

Прерывания, возникающие при работе вычислительной машины, можно разделить на несколько типов (рис.2.30).

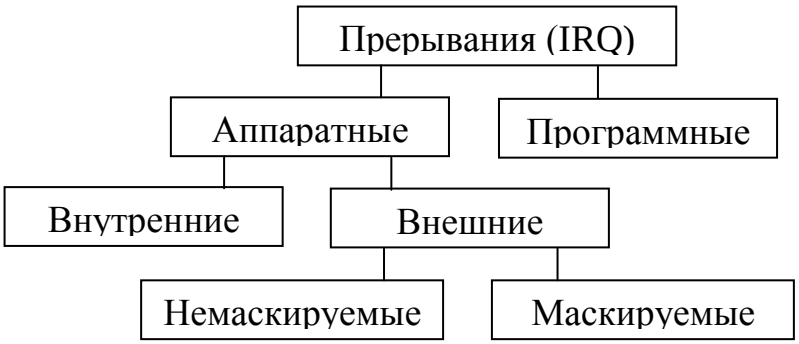


Рис. 2.30. Классификация типов прерываний

Аппаратные прерывания вызываются физическими устройствами и возникают по отношению к программе асинхронно, т.е. в общем

случае невозможно предсказать, когда и по какой причине программа будет прервана.

Аппаратные прерывания не координируются с работой программного обеспечения. Когда вызывается прерывание, то процессор прерывает свою работу, выполняет прерывание, а затем возвращается на прежнее место.

Внешние прерывания возникают по сигналу какого-либо внешнего устройства, например:

- прерывание, которое информирует систему о том, что требуемый сектор диска уже прочитан, его содержимое доступно программе;
- прерывание, которое информирует ситему о том, что завершилась печать символа на принтере и необходимо выдать следующий символ;
- прерывания по нарушению питания;
- нормальное завершение некоторой операции ввода-вывода (нажатие клавиши на клавиатуре);
- прерывание по таймеру.

Внутренние прерывания вызываются событиями, которые связаны с работой процессора и являются синхронными с его операциями, а именно прерываение происходит:

при нарушении адресации (в адресной части выполняемой команды указан запрещенный или несуществующий адрес, обращение к отсутствующему сегменту или странице при организации механизмов виртуальной памяти);

- при наличии в поле кода незадействованной двоичной комбинации;
- при делении на ноль;
- при переполнении или исчезновении порядка;
- при обнаружении ошибок четности, ошибок в работе различных утсройство аппаратуры срдствами контроля.

Запросы, возникающие в таких (исключительных) ситуациях, называются **исключениями.**

Исключения делятся на **отказы** и **выходы из процесса**, в зависимости от способа сообщения о них и возможности перезапуска процессора с вызвавшей их команды. **Отказы** – это исключения, которые выявляются и обслуживаются перед выполнением команды. Они могут иметь место в виртуальной системе памяти, когда процессор обращается к несуществующим странице или сегменту. В процессе обработки такого исключения операционная система обращается к странице или сегменту на диске, а процессор перезапускает команду. **Выход из процесса** является исключением, которое не позволяет точно локализовать

причину, вызвавшую исключительную ситуацию. Выходы из процесса используются для сообщения о крупных ошибках, таких как неисправности аппаратуры или ошибки в системных таблицах. Адресом возврата из процедуры обработки исключений всегда является команда, вызвавшая исключение, возможны префиксы.

Программные прерывания не являются асинхронными. Программы могут сами вызывать прерывания с заданным номером. Для этого они используют команду INT. По этой команде процессор осуществляет практически те же действия, что и при обычных прерываниях, но только это происходит в предсказуемой точке программы. Там, где программист поместил данную команду.

Программные прерывания в прямом смысле прерываниями не являются, поскольку представляют собой лишь специфический способ вызова процедур не по адресу, а номеру в таблице.

Механизм программных прерываний был специально введен для того, чтобы:

- переключение на системные программные модули происходило не просто как переход в подпрограмму, а точно таким же образом, как и обычные прерывания. Этим обеспечивается автоматическое переключение процессора в привилегированный режим с возможностью исполнения любых коман;
- использование программных прерываний приводит к более компактному коду программ по сравнению с использованием стандартных команд выполнения процедур.

В дальнейшем обработку всех запросов, прерывающих ход выполнения текущей программы, будем называть просто **прерываниями.**

Программу, затребованную запросом прерывания, назовем **прерывающей программой,** противопоставляя ее **прерываемой программе,** выполнявшейся в ЭВМ до появления запроса.

Характеристики системы прерывания

Для оценки эффективности систем прерывания могут быть использованы следующие характеристики:

1. **Общее число запросов прерывания** (входов в систему прерывания).

2. Время реакции – время между появлением запроса прерывания и моментом прерывания текущей программы. На рис. 2.31 приведена упрощенная временная диаграмма процесса прерывания.

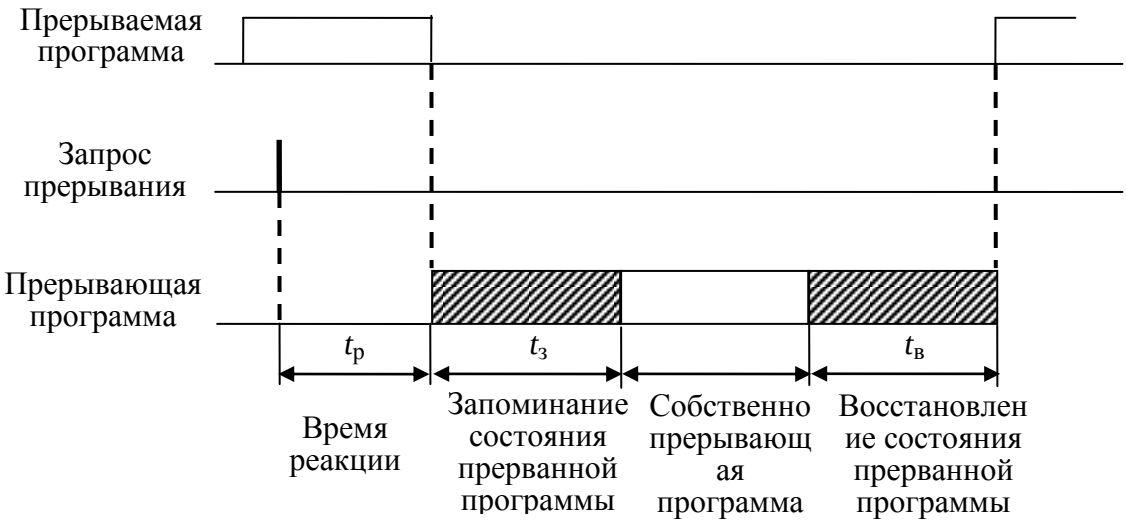


Рис. 2.31. Упрощенная временная диаграмма процесса прерывания

Для одного и того же запроса задержки в исполнении прерывающей программы зависят от того, сколько программ со старшим приоритетом ждут обслуживания, поэтому время реакции определяют для запроса с наивысшим приоритетом ($t_Р$).

Время реакции зависит от того, в какой момент допустимо прерывание. Большей частью прерывание допускается после окончания текущей команды. В этом случае время реакции определяется в основном длительностью выполнения команды.

Это время реакции может оказаться недопустимо большим для ЭВМ, предназначенных для работы в реальном масштабе времени. В таких машинах часто допускается прерывание после любого такта выполнения команды (микрокоманды). Однако при этом возрастает количество информации, подлежащей запоминанию и восстановлению при переключении программ, т.к. в этом случае необходимо сохранять также и состояние счетчика тактов, регистра кода операции и некоторых других узлов. Такая организация прерывания возможна только в машинах с быстродействующей сверхоперативной памятью.

Имеются ситуации, в которых желательно немедленное прерывание. Если аппаратура контроля обнаружила ошибку, то целесообразно сразу же прервать операцию, пока ошибка не оказала влияние на следующие такты работы программы.

3. Затраты времени на переключение программ (издержки прерывания) равны суммарному расходу времени на запоминание и восстановление состояния программы (рис. 2.31):

$$t_{изд} = t_з + t_в.$$

4. Глубина прерывания – максимальное число программ, которые могут прерывать друг друга. На рис. 2.32 показаны процессы прерывания в системах с различной глубиной прерывания (предполагается, что приоритет каждого последующего запроса выше предыдущего). Если после перехода к прерывающей программе и вплоть до ее окончания прием запросов прекращается, то говорят, что система имеет глубину прерывания, равную 1 (относительная дисциплина обслуживания).

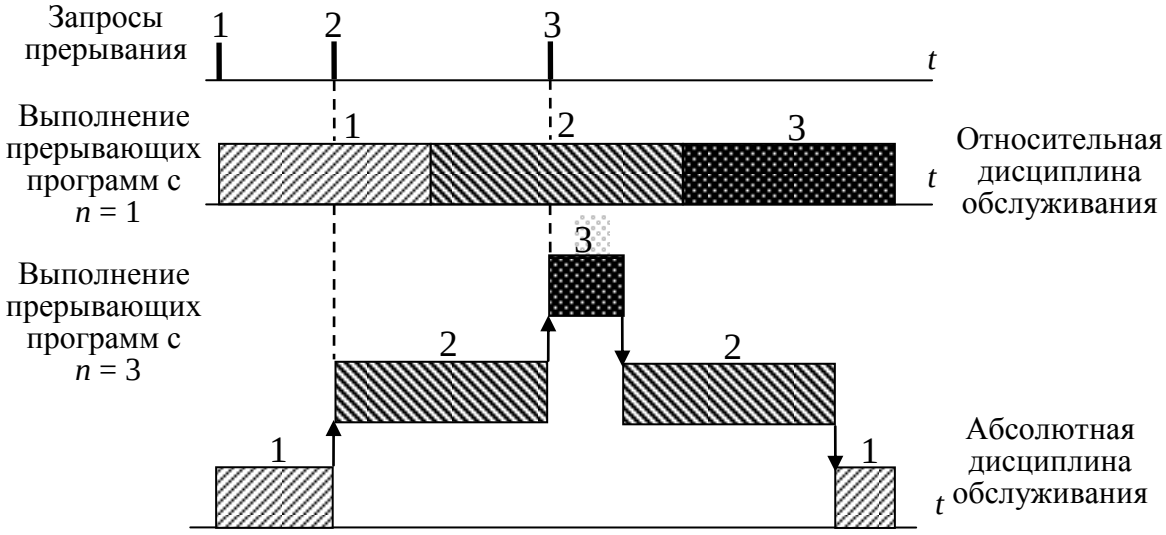


Рис. 2.32. Процессы прерывания с различной глубиной прерывания и дисциплиной обслуживания

Глубина равна *n*, если допускается последовательное прерывание до *n* прерывающих программ (абсолютная дисциплина обслуживания). Глубина прерывания обычно совпадает с числом уровней приоритета в системе прерывания. Система с большим значением глубины прерывания обеспечивает более быструю реакцию на срочные запросы.

Если запрос окажется не обслуженным к моменту прихода нового запроса от того же источника, то возникает так называемое **насыщение системы прерывания.** В этом случае предыдущий запрос от данного источника будет машинально утерян, что недопустимо. Существуют специальные методы борьбы с этой ситуацией.

Программно-управляемый приоритет прерывающих программ

Относительная степень важности программ, их частота повторения, относительная степень срочности в ходе вычислительного процесса могут меняться, требуя установления новых приоритетных отношений. Поэтому во многих случаях приоритет между прерывающими программами не может быть зафиксирован раз и навсегда. Необходимо иметь возможность изменять, по мере необходимости, приоритетные соотношения программным путем. Приоритет между прерывающими программами должен быть динамичным, т.е. программно управляемым.

В ЭВМ широко применяется способ маскирования прерываний.

Существуюбт два специальных внешних сигнала среди входных сигналов процессора, при помощи которых можно прервать выполнение текущей программы и тем самым переключить работу центрального процессора. Это сигналы NMI (Non Mascable Interrupt, немаскируемое прерывание) INTR (INTerrupt Request, запрос на прерывание).

Соответственно внешние прерывания подразделяются на два вида: немаскируемые и маскируемые.

Маска прерывания представляет собой двоичный код, разряды которого поставлены в соответствие запросам или классам (уровням) прерывания. Маска загружается командой программы в регистр маски (рис. 2.33).

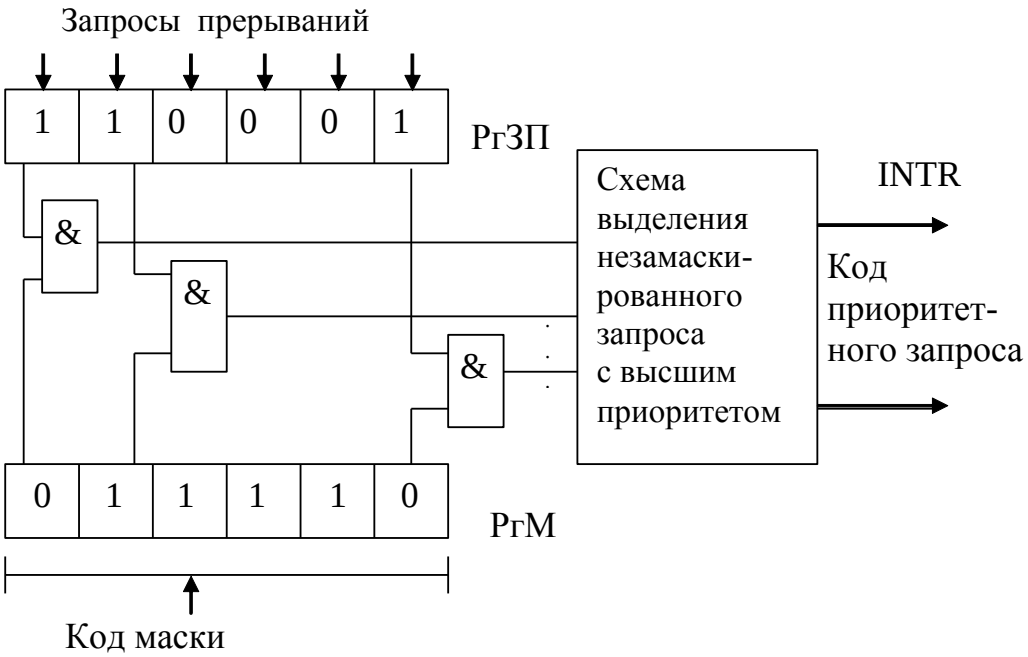


Рис. 2.33. Маскирование прерываний

Состояние 1 в данном разряде регистра маски (РгМ) разрешает, а состояние 0 запрещает (маскирует) прерывание текущей программы от соответствующего запроса в регистре запросов прерываний (РгЗП). Таким образом, программа, изменяя маску в регистре маски, может устанавливать произвольные приоритетные соотношения между программами без перекоммутации линий, по которым поступают запросы прерывания. Каждая прерывающая программа может установить свою маску. При формировании маски 1 устанавливаются в разряды, соответствующие запросам (прерывающим программам) с более высоким, чем у данной программы, приоритетом. Схемы И выделяют поступившие незамаскированные запросы прерывания, из которых специальная схема выделяет наиболее приоритетный запрос, формирует код его номера и вырабатывает основной сигнал прерывания (INTR).

С замаскированным запросом, в зависимости от причины прерывания, поступают двояким образом: или он игнорируется, или запоминается, с тем, чтобы осуществить затребованные действия, когда запрет будет снят. Например, если прерывание вызвано окончанием операции в ПУ, то его следует, как правило, запомнить, так как иначе ЭВМ останется неосведомленной о том, что ПУ освободилось.

Возможность прерывания программ – важное архитектурное свойство ЭВМ, позволяющее эффективно использовать производительность процессора при наличии нескольких процессов, протекающих параллельно во времени, требующих в произвольные моменты времени управления и обслуживания со стороны процессора.

Чтобы ЭВМ могла, не требуя больших усилий от программиста, реализовывать с высоким быстродействием прерывания программ, машине необходимо придать соответствующие аппаратурные и программные средства, совокупность которых получила название **системы прерывания программ.** В качестве аппаратных средств используется **контроллер прерывания** (блок прерывания).

Основными функциями системы прерывания являются:

- запоминание состояния прерываемой программы и осуществление перехода к прерывающей программе;
- восстановление состояния прерванной программы и возврат к ней.

При наличии нескольких источников запросов прерывания между ними должны быть установлены **приоритетные соотношения,** определяющие, какой из нескольких поступивших запросов подлежит обработке в первую очередь, и **устанавливающие**: имеет право или нет

данный запрос (прерывающая программа) прерывать ту или иную программу.

Структура аппаратных средств системы прерывания

В процессе эволюции компьютеров на базе Intel, эволюционировала и архитектура обработки прерываний. Изначально контроллер прерываний **PIC (Programmable Interrupt Controller)** состоял из одной микросхемы Intel 8259 с 8-ю входными линиями от устройств и двумя выходными сигнальными линиями к процессору — INT и NMI. Кроме того контроллер прерываний связан с процессором через шину. Контроллер вызывает внимание процессора выставлением сигнала INT или NMI, а номер возникшего прерывания передает через шину. Кроме того, через шину процессор может осуществлять управление контроллером. На физическом уровне различают даже несколько шин, через которые связаны контроллер с процессором.

Восемь входных линий контроллера прерываний позволяли назначать устройствам 8 разных аппаратных прерываний. Позже этого оказалось не достаточно. Чтобы расширить количество входных линий в IRQ два одинаковых контроллера связали каскадно. Выходная сигнальная линия дополнительного контроллера завязывалась на входную линию главного контроллера с номером 2. Таким образом, для устройств в такой совместной схеме осталось 15 входных линий.

С появлением многоядерных систем потребовались дополнительные контроллеры прерываний. Интелу потребовалось решать две задачи. Во-первых, снова появилась нехватка входных линий прерываний. Во-вторых, потребовался механизм межпроцессорной синхронизации в многопроцессорных системах. На смену PIC, Intel разработал APIC.

APIC (Advanced Programmable Interrupt Controller) — улучшенный программируемый контроллер прерываний.

Преимущества расширенного контроллера прерываний:

- возможность реализации межпроцессорных прерываний — сигналов от одного процессора другому;
- поддержка до 256 входов IRQ, в отличие от 1 5 в классической архитектуре;
- крайне быстрый доступ к регистрам текущего приоритета прерывания и подтверждения прерывания.

APIC поддерживался в ОС Windows, начиная с Windows NT 4.0.

APIC состоит из двух модулей:

- **LOCAL APIC** — располагается в ядре процессора, если система многоядерна - в каждом ядре;

- **I/O APIC** — контроллер, расположенный на системной плате, обычно как часть микросхем обрамления.

На примере структуры соединения контроллеров прерывания (рис. 2.34) внутрь каждого ядра процессора поместили контроллер с именем Local APIC, а в дополнение к 2-м контроллерам 8259, добавили IOAPIC. Появилась дополнительная шина APIC, связывающая между собой IOAPIC и LocalAPIC. Появились линии синхронизации процессоров.

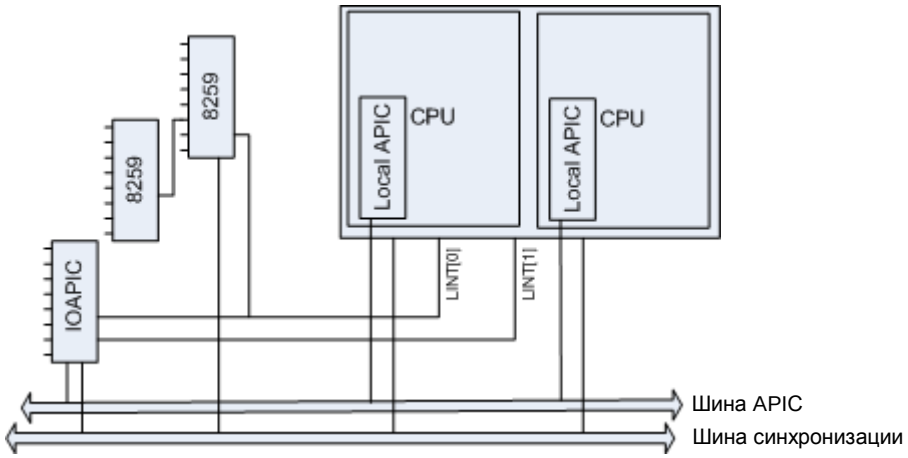


Рис. 2.34. Пример структуры соединия контроллеров прерывания

В настоящий момент наблюдается тенденция к отказу от IO APIC, как и проводников IRQ, и переходу на Message Signaled Interrupts.

Message Signaled Interrupts (MSI – прерывания, инициируемые сообщениями) — альтернативная форма прерываний, доступная в PCI версии 2.2 и более поздних, PCI-X, а также обязательная в PCI Express любых версий. Вместо присваивания фиксированного номера запроса на прерывание, устройству разрешается записывать сообщение по определённому адресу системной памяти, на деле отображенному на аппаратуру локального контроллера прерываний (local APIC) процессора. Для записи сообщения используется тот же механизм захвата шины (bus mastering), что и для DMA (Direct Memory Access – прямой доступ к памяти).

Для записи сообщений каждое устройство, использующее MSI может иметь от одной до тридцати двух уникальных областей памяти.

Все прерывания шины PCI Express всегда доставляются как MSI, даже при использовании эмуляции традиционных номеров проводников прерываний.

Достоинства MSI:

• возможность полного отказа от проводников IRQ от устройств и разъемов PCI до главного контроллера прерываний (IO APIC), а также от самого главного контроллера прерываний, что упрощает материнскую плату;

• в многопроцессорных и многоядерных системах устройства, использующие несколько областей MSI, получают возможность самостоятельно выбирать процессор/ядро для обработки конкретного прерывания, причем делать это полностью на уровне аппаратуры без исполнения программного кода. Это позволяет оптимизировать работу путем размещения большей части структур драйвера устройства и связанного с ним программного обеспечения (сетевых протоколов и т.д.) в кэше конкретного процессора.

MSI поддерживается в операционных системах Microsoft Windows Vista и более поздних, в ОС FreeBSD с версии 6.3, а также в ядре Linux начиная с версии 2.6.8.

Организация перехода к прерывающей программе

Наиболее гибким и динамичным является **векторное прерывание,** при котором вектор начального состояния прерывающей программы называют **вектором прерывания.** Он содержит всю необходимую информацию для перехода к прерывающей программе, в том числе ее начальный адрес. Каждому запросу (уровню) прерывания соответствует свой вектор прерывания, способный инициировать выполнение соответствующей прерывающей программы. Векторы прерывания обычно находятся в специально выделенных фиксированных ячейках памяти (стеке).

Главное место в процедуре перехода к прерывающей программе занимает передача из соответствующего регистра (регистров) процессора в память (стек) на сохранение текущего вектора состояния прерываемой программы (чтобы можно было вернуться к ее исполнению) и загрузка в регистр (регистры) процессора вектора прерывания прерывающей программы, к которой при этом переходит управление процессором. Источник прерывания, выставляя запрос прерывания, посылает в процессор (выставляет на шины интерфейса) код адреса в памяти своего вектора прерывания.

При векторном прерывании каждому запросу прерывания, или, другими словами, устройству – источнику прерывания, соответствует переход к начальному адресу соответствующей прерывающей программы, задаваемому вектором прерывания.

3. ФУНКЦИОНАЛЬНАЯ И СТРУКТУРНАЯ ОРГАНИЗАЦИЯ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА ЭВМ

В области вычислительной техники различают процессоры центральные, графические, ввода/вывода, передачи данных, коммуникационные и специализированные.

3.1. Назначение и структура центрального процессора

Центральный процессор – основное устройство ЭВМ, которое наряду с обработкой данных выполняет функции управления системой: инициирование ввода/вывода, обработку системных событий, управление доступом к основной памяти и т.п.

Структурная организация центрального процессора (ЦП) определяется функционально-логической организацией, микроархитектурой и требованиями к технико-экономическим показателям.

Логическую структуру ЦП представляет ряд функциональных средств (рис. 3.1): средства обработки, средства управления системой и программой (центральное устройство управления), локальная память, буферная память (кэш-память L1, L2, …), средства инициализации ввода/вывода, средства управления памятью, системные средства.

Средства обработки обеспечивают выполнение операций над данными с фиксированной (целочисленные данные) и плавающей точкой, векторными данными, полями переменной длины и т.д. Локальная память состоит из регистров общего назначения, регистров данных с плавающей точкой, векторных регистров, управляющих регистров и др. К средствам управления памятью относятся средства управления доступом к ОП и предвыборкой команд и данных. Буферная память включает в себя кэш-память команд и данных первого (L1), второго (L2) и третьего (L3) уровней. Средства инициализации ввода/вывода обеспечивают активизацию контроллеров (каналов) периферийных устройств. К системным средствам относятся средства службы времени: часы астрономического времени, таймер, коммутатор и т.д. Существует обязательный (стандартный) минимальный набор функциональных средств для каждого типа центрального процессора. Он включает в себя регистры общего назначения, средства выполнения стандартного набора операций и средства управления вычислительным процессом.

Рис. 3.1. Логическая организация ЦП

3.2. Назначение, классификация и организация ЦУУ

Центральное устройство управления (ЦУУ) – это комплекс средств автоматического управления процессом передачи и обработки информации. ЦУУ вырабатывает управляющие сигналы (УС), необходимые для выполнения всех операций, предусмотренных системой команд, а также координирует работу всех узлов и блоков ЭВМ. В связи с этим можно считать ЦУУ преобразователем первичной командной информации, представленной программой решения задачи, во вторичную командную информацию, представляемую управляющими сигналами.

В общем случае ЦУУ формирует управляющие сигналы для реализации следующих функций:

- выборки из памяти кода очередной команды;
- расшифровки кода операции и признаков выбранной команды;
- выборки операндов и выполнения машинной операции;
- обеспечения прерываний при выполнении команд;
- формирования адреса следующей команды;
- учета состояний других устройств машины;
- инициализации работы контроллеров (каналов) ввода/вывода;
- организации контроля работоспособности ЭВМ.

По общей организации управление может быть центральным, распределенным и смешанным. В первом случае в блоке управления ЦУУ вырабатываются все УС для всех команд, выполняемых процессором ЭВМ. Во втором случае операционные и другие устройства процессора имеют собственные блоки местного управления. В последнем случае ЦУУ вырабатывает сигналы для запуска в работу блоков местного управления.

По способу синхронизации работы различают ЦУУ:

- синхронного типа, в которых время цикла может быть постоянным или переменным;
- асинхронного типа, в которых продолжительность цикла определяется фактическими затратами времени на выполнение каждой операции, в этом случае необходимо вырабатывать сигналы об окончании операции;
- смешанного типа, где частично реализуются оба предыдущих принципа организации циклов.

По принципу формирования и развертывания временной последовательности УС различают ЦУУ:

- аппаратного (схемного) типа;
- микропрограммного типа.

Центральное устройство управления микропрограммного типа

Микропрограммный принцип управления обеспечивает реализацию одной сложной машинной команды путем выполнения определенной микропрограммы, интерпретирующей алгоритм выполнения данной операции. Совокупность микропрограмм, необходимая для реализации сложных команд ЭВМ, хранится в специальной памяти микропрограмм. Каждая микропрограмма состоит из определенной последовательности микрокоманд, которые после выборки из памяти преобразуются в набор управляющих сигналов (технология Micro-ops fusion от Intel).

Анализ аппаратурной (схемной) и микропрограммной реализации устройства управления указывает на зависимость стоимости управления от сложности выполняемых команд. Для простых команд выгодно использовать схемное управление, а для сложных команд – микропрограммное. Однако последнее приводит к увеличению затрат времени на выработку управляющих воздействий. Основным же преимуществом микропрограммного управления является его гибкость, которая позволяет повышать эффективность серийно выпускаемых и эксплуатируемых машин за счет введения новых средств математического обеспечения, использующих дополнительный набор команд и новые функции процессора. Модернизация алгоритмов или реализация дополнительных команд легко осуществляется путем изменения содержимого микропрограммной памяти.

3.3. Регистровые структуры центрального процессора

3.3.1. Регистровые структуры процессоров IA-32

В процессорах IA-32 можно выделить следующие группы регистров:

1. Основные функциональные регистры:

- регистры общего назначения (GPR);
- указатель команд;
- регистр флагов;
- регистры сегментов.

2. Регистры процессора обработки чисел с плавающей точкой (FPU):

- регистры данных;
- регистр тегов;
- регистр состояния;
- регистр указателей команд и данных FPU;

- регистр управления FPU.

3. Векторные регистры расширений SSE.

4. Системные регистры:

- регистры управления микропроцессора;
- регистры системных адресов.

5. Регистры отладки и тестирования.

Регистры первых трех групп используются при выполнении прикладных программ, четвертой группы – системных операций, пятой – при отладке и тестировании.

Регистры общего назначения

Восемь 32-разрядных регистров (EAX, ECX, EDX, EBS, EBP, ESP, ESI, EDI) предназначены для хранения данных и адресов. Они поддерживают работу с данными разрядностью 1, 8, 16 и 32 бита, битовыми полями длиной от 1 до 32 бит и адресами размером 16 и 32 бита. Младшие 16 разрядов этих регистров (рис.3.2) доступны отдельно при использовании соответствующего имени, например регистр EAX (имя AX для 16 разрядов).

При операциях с байтами можно отдельно обращаться к младшему байту (разряды 0–7) и старшему байту (8–15) по именам AL и AH. Доступ к отдельным байтам обеспечивает дополнительную гибкость при операциях с данными.

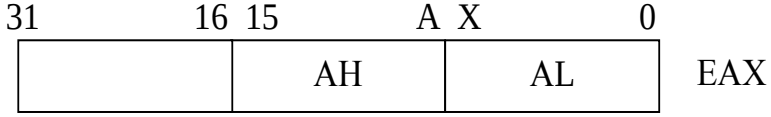


Рис. 3.2. Структура регистра общего назначения EAX

Регистры сегментов и дескрипторы сегментов

Шесть 16-разрядных сегментных регистров (CS, SS, DS, ES, FS, GS) содержат значения селекторов сегментов, указывающих на текущие адресуемые сегменты памяти. С каждым из них связан программно-недоступный регистр дескриптора сегмента (рис. 3.3).

В защищенном режиме каждый сегмент может иметь размер от 1 байта до 4 Гбайт, в режиме реальных адресов максимальный размер сегмента составляет 64 Кбайта.

Селектор в CS обеспечивает обращение к текущему сегменту команд, селектор в SS – к текущему сегменту стека, селекторы в DS, ES,

FS, GS – к текущим сегментам данных. Каждый регистр дескриптора содержит базовый адрес сегмента, 32-разрядный размер сегмента и другие необходимые атрибуты.

Рис. 3.3. Регистры сегментов и соответствующие регистры дескрипторов

Когда в регистр сегмента загружается новое значение селектора, содержимое соответствующего регистра дескриптора автоматически корректируется. В реальном режиме базовый адрес сегмента получается путем сдвига значения селектора на 4 разряда влево (20 разрядов), максимальный размер и атрибуты сегмента в реальном режиме имеют фиксированные значения.

Указатель команд

Указатель команд (рис. 3.4) представляет собой 32-разрядный регистр с именем EIP, содержимое которого используется в качестве смещения при определении адреса следующей выполняемой команды. Смещение задается относительно базового адреса сегмента команд CS. Младшие 16 бит (0–15) содержат 16-разрядный указатель команд с именем IP, который используется при 16-разрядной адресации.

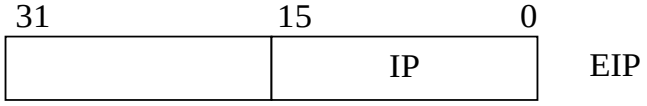


Рис. 3.4. Структура регистра указателя команд

Указатель команд непосредственно программисту недоступен. Его содержимое изменяется при выполнении команд передачи управления и прерываний.

Регистр флагов

Регистр флагов является 32-разрядным, имеет имя EFLAGS. Его разряды содержат признаки результата выполнения команды, управляют обработкой прерываний, последовательностью вызываемых задач, вводом/выводом и рядом других процедур.

Регистры процессора обработки чисел с плавающей точкой

Набор регистров, входящих в блок FPU, изображен на рис. 3.5.

При работе FPU 80-разрядные регистры ST0–ST7 образуют кольцевой стек, в котором хранятся числа с плавающей точкой, представленные в формате с расширенной точностью.

Рис. 3.5. Регистры блока FPU

Регистр тегов FPU содержит 16-разрядное слово, включающее восемь двухбитовых тегов. Каждый тег (признак) характеризует содержимое одного из регистров данных.

Тег определяет, является ли регистр пустым (незаполненным) – код 11, или в него введено конечное число – 00 (достоверное значение), или нуль – 01, неопределенное значение (бесконечность) – 10 (нет числа

и неподдерживаемый формат). Слово тегов позволяет оптимизировать функционирование FPU посредством идентификации пустых и непустых регистров данных, проверить содержимое регистра без сложного декодирования хранящихся в нем данных.

Регистры MMX-технологии

При реализации MMX-команд регистры данных FPU используются как 64-разрядные регистры MM0–MM7 (см. рис. 3.5), где могут храниться несколько целочисленных операндов (восемь 8-разрядных, четыре 16-разрядных, два 32-разрядных или один 64-разрядный), над которыми одновременно выполняется поступившая в процессор команда.

Векторные регистры SSE расширений

Потоковые команды расширений SSE используют восемь 128-разрядных регистров XMM0–XMM7, в которых могут храниться несколько вещественных или целочисленных операндов.

Системные регистры

Системные регистры управляют функционированием микропроцессора в целом и режимами работы отдельных внутренних блоков: процессора с плавающей точкой, кэш-памятью, диспетчера памяти. Эти регистры доступны только в защищенном режиме для программ.

Набор системных регистров включает регистры управления, регистры системных адресов и сегментов.

Регистры управления 32-разрядные, служат для фиксации общего состояния процессора. Эти регистры вместе с регистрами системных адресов хранят информацию о состоянии процессора, которое затрагивает все задачи.

Регистры отладки и тестирования

Набор программно-доступных регистров поддерживает отладку программ и тестирования внутренних блоков процессора. Встроенный алгоритм самотестирования (BIST) осуществляет поиск ошибки в микрокоде и в больших логических матрицах, а также тестирование кэш-памяти команд и данных, буферов ассоциативной трансляции (TLB) и устройств постоянной памяти. Внутренние счетчики контролируют работу процессора и проводят подсчет событий. Введена новая функция – мониторинг термического состояния системной платы.

Переименование регистров

В современных процессорах используются блоки регистров замещения (регистровые файлы) для целочисленных, вещественных и векторных данных. Для любого указанного в команде логического регистра (программно можно обращаться в x86 только к 8-ми регистрам общего назначения GPR, 8-ми регистрам с плавающей точкой ST или 8-ми MMX / XMM-регистрам) выделяется один из физических регистров соответствующего блока регистров замещения, содержащего, например, 128 регистров. Эта процедура (переименование регистров) позволяет увеличить количество используемых регистров процессора, а также позволяет выполнять команды, в которых задействованы одни и те же логические регистры, одновременно или с изменением их последовательности.

3.3.2. Регистровые структуры процессоров AMD64 (Intel64)

В процессорах x86-64 (AMD64), Intel64 архитектур (рис. 3.6) существующие в x86 регистры общего назначения (GPR) расширены с 32 до 64 бит (RAX, RBX, RCX, RDX, RBP, RSP, RSI, RDI) и к ним добавлены еще 8 новых 64-разрядных регистров (R8–R15). Также 8 новых 128-битных регистров (XMM8–XMM15) добавлено в блок SSE, что обеспечивает поддержку SSE2.

В блоке FPU используются существующие в x87 регистры данных ST0–ST7 (80-разрядные) и 64-разрядные мультимедийные регистры MM0–MM7, объединенные в общее пространство с регистрами ST.

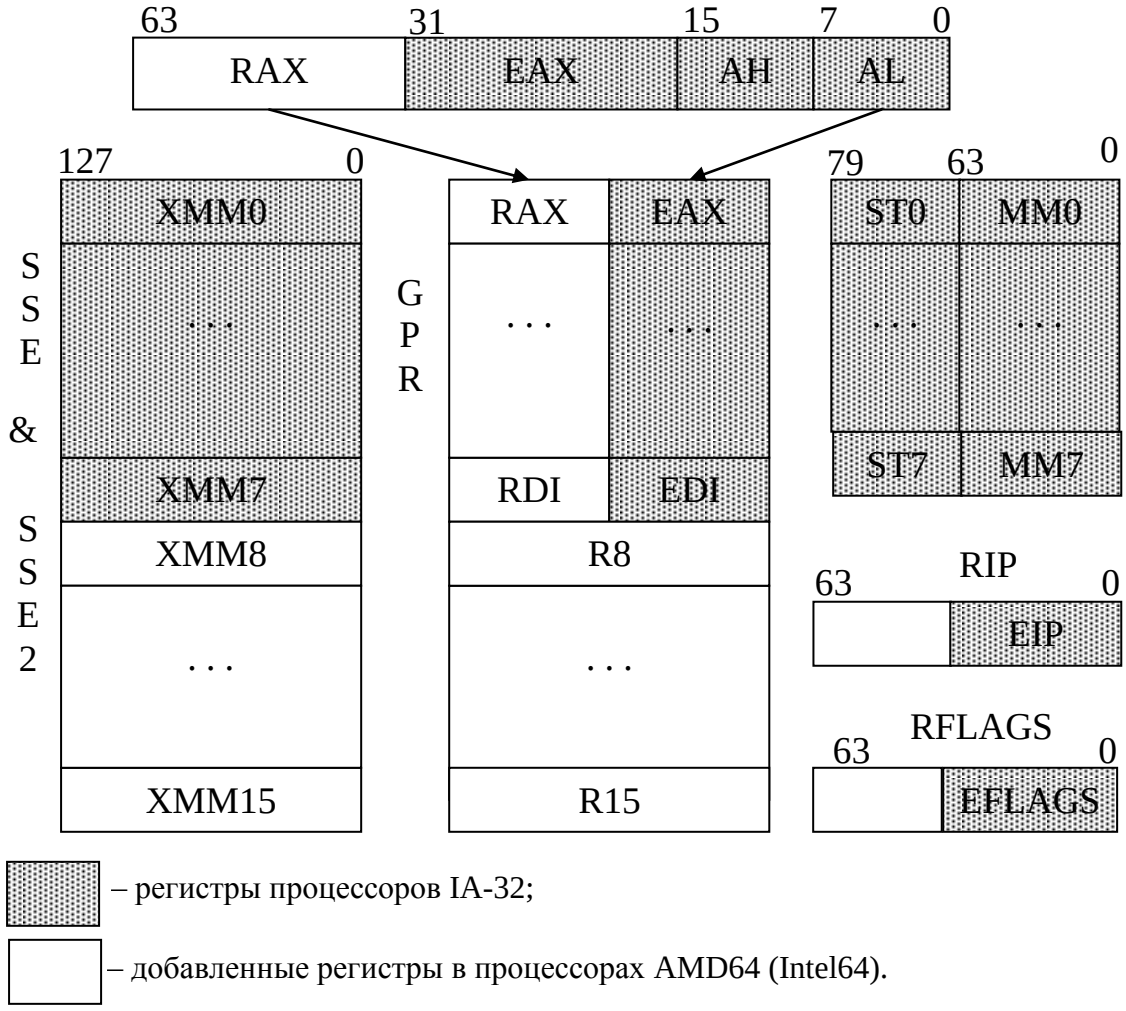
Регистр указателя команд (RIP) и регистр флагов (RFLAGS) также расширены до 64 разрядов.

Векторные регистры AVX расширений

Инструкции расширений AVX, AVX2 используют шестнадцать 256-битных регистров YMM0–YMM15, в которых могут храниться несколько целочисленных или с плавающей запятой опернадов.

Векторные регистры MIC расширений

Инструкции расширений MIC используют шестнадцать 512-битных регистров, в которых могут храниться несколько целочисленных или с плавающей запятой опернадов.



– регистры процессоров IA-32;

– добавленные регистры в процессорах AMD64 (Intel64).

Рис. 3.6. Регистровые структуры процессоров AMD64 (Intel64)

3.3.3. Регистровые структуры процессоров IA-64

В состав регистровых файлов IA-64 (рис.3.7) входят: 128 регистров общего назначения GPR (64-разрядных); 128 регистров с плавающей запятой FR (82-разрядных); 128 прикладных регистров (в основном 64-разрядных) AR; 64 одноразрядных регистра предикатов PR; 8 регистров переходов BR (64-разрядных); не менее 4-х регистров идентификатора процесса CPUID; счетчик команд IP; регистр маркера текущего окна CFM стека регистров и др.

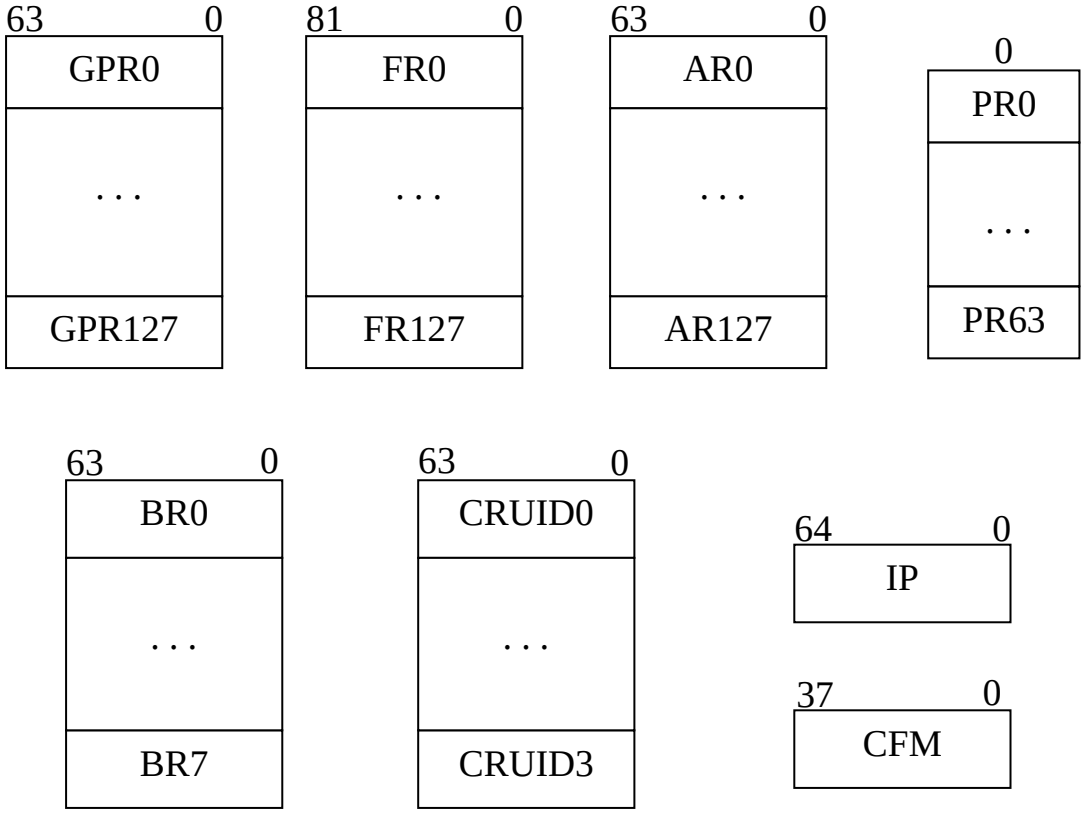


Рис. 3.7. Регистровые структуры процессоров IA-64

64-разрядные регистры GPR0–GPR127 применяются не только для целочисленных операций IA-64; GPR8–GPR31 в режиме IA-32 используются также под целочисленные регистры и регистры селекторов и дескрипторов сегментов IA-32. GPR0–GPR31 называются **статическими регистрами** (GPR0 всегда содержит 0), а GPR32–GPR127 – **стекируемыми регистрами**. Статические регистры «видны» всем программам. Стекируемые регистры становятся доступными в программной единице через окно стека регистров, включающее локальные и выходные регистры, число которых задается командой alloc.

82-разрядные регистры с плавающей запятой FR0–FR127 также подразделяются на **статические** (FR0–FR31, причем всегда FR0 = 0.0, FR1 = 1.0) и **вращаемые** (FR32–FR127). FR8–FR31 в режиме IA-32 содержат числа с плавающей запятой и мультимедийные регистры. Вращение регистров является в некотором роде частным случаем переименования регистров, применяемым в современных суперскалярных процессорах с внеочередным выполнением команд. В отличие от них (пе-

реименование регистров осуществляется аппаратно) вращение регистров в IA-64 управляется программно.

Прикладные регистры AR0–AR127 – специализированные. Ряд AR-регистров является фактически регистрами IA-32; AR0–AR7 называются регистрами ядра. Запись в них привилегированна, но они доступны на чтение в любом приложении и используются для передачи приложению сообщений от операционной системы. AR16 (RSC) – регистр конфигурации стека регистров, используемый для управления работой стека регистров IA-64; AR40 (FPSR) – регистр состояния для команд с плавающей запятой IA-64.

Регистры предикатов PR0–PR63 являются одноразрядными, в них помещаются результаты выполнения команд сравнения. Обычно эти команды устанавливают сразу два соседних регистра PR в состояния «1» – истина, «0» – ложь или, наоборот, в зависимости от значения условия. Такая избыточность обеспечивает дополнительную гибкость.

64-разрядные регистры переходов BR0–BR7 применяются для указания адреса перехода в соответствующих командах перехода (если адрес перехода не кодируется в команде явно).

В регистрах CPUID 0 и CPUID 1 находится информация о производителе, в регистре CPUID 2 – серийный номер процессора, а в регистре CPUID 3 задается тип процессора (семейство, модель, версия архитектуры и т.п.) и число CPUID-регистров. Разряды регистра CPUID4 указывают на поддержку конкретных особенностей IA-64, которые реализованы в данном процессоре.

3.4. Структурная организация современных универсальных микропроцессоров

Характерными чертами современных универсальных микропроцессоров являются:

1. Суперскалярная архитектура, обеспечивающая одновременное выполнение нескольких команд в параллельно работающих исполнительных устройствах.
2. Динамическое изменение последовательности команд (выполнение команд с опережением – спекулятивное выполнение).
3. Конвейерное исполнение команд.
4. Предсказание направления ветвлений.
5. Предварительная выборка команд и данных.
6. Параллельная обработка потоков данных.
7. Многоядерная структура.

8. Многопотоковая обработка команд.

9. Пониженное энергопотребление.

Практическая реализация данных принципов в структурах различных процессоров имеет ряд существенных особенностей, связанных с их микроархитектурой. **Микроархитектура** процессора определяет реализацию его внутренней структуры, принципы выполнения поступающих команд, способы размещения и обработки данных.

3.4.1. Стратегия развития процессоров Intel

Стратегия развития Intel заключается во внедрении новых микроархитектур процессоров, основанных на новых поколениях полупроводниковой производственной технологии. Темпы выпуска инновационных микроархитектур и полупроводниковых технологий основаны на принципе, который корпорация Intel называет моделью «TICK-TOCK» («ТИК-ТАК»). Каждый «TICK» обозначает (табл. 3.1) новый этап развития полупроводниковых технологий (техпроцесс – 65 нм, 45 нм, 32 нм, 22 нм, 14 нм, 8 нм и т.д.), а каждый «TOCK» – создание новой микроархитектуры (Intel Core, Nehalem, Sandy Bridge, Haswell, Skylake). Переход на новый техпроцесс сопровождается выпуском соответствующих семейств процессоров (Penryn, Westmere, Ivy Bridge, Broadwell, Skymont).

Таблица 3.1

Стратегия развития процессоров Intel

Intel Core	Новая микроархитектура	65 нм	2007 г.	TOCK
Intel Penryn	Семейство процессоров с новым техпроцессом	45 нм	2008 г.	TICK
Intel Nehalem	Новая микроархитектура	45 нм	2009 г.	TOCK
Intel Westmere	Семейство процессоров с новым техпроцессом	32 нм	2010 г.	TICK
Intel Sandy Bridge	Новая микроархитектура	32 нм	2011 г.	TOCK
Intel Ivy Bridge	Семейство процессоров с новым техпроцессом	22 нм	2012 г.	TICK
Intel Haswell	Новая микроархитектура	22 нм	2013 г.	TOCK
Intel Broadwell	Семейство процессоров с новым техпроцессом	14 нм	2014 г.	TICK
Intel Skylake	Новая микроархитектура	14 нм	2015 г.	TOCK
Intel Skymont	Семейство процессоров с новым техпроцессом	10 нм	2016 г.	TICK

Этот цикл, как правило, повторяется каждые 2 года. Новаторская микроархитектура «обкатывается» на текущем производственном про-

цессе, затем переносится на новую производственную технологию. Данная модель развития позволяет осуществлять внедрение единообразной процессорной микроархитектуры во всех сегментах рынка.

Стратегия развития архитектуры и полупроводниковой технологии, реализуемая корпорацией Intel, не только позволяет выпускать новые решения в соответствии с запланированными темпами, но и способствует внедрению инновационных решений в отрасли на уровне платформ, расширяя использование преимуществ высокой производительности и энергоэкономичности.

3.4.2. Микроархитектура Intel Core

Рассмотрение различных микроархитектур процессоров Intel начнем с микроархитектуры Intel Core. Вследствие того, что она получилась очень удачной и основные нововведения, используемые в ней, получили дальнейшее развитие в других микроархитектурах, появившихся позже. Микроархитектура Intel Core наследует философию эффективного энергопотребления, впервые реализованную в процессорах Intel Pentium M для мобильных ПК. Заимствовав лучшее от ставших основой для настольных и мобильных процессоров Intel микроархитектур Net Burst и Mobile, микроархитектура Intel Core содержало сотни нововведений, но основные из них сводятся к пяти технологическим решениям:

1. **Технология Intel Wide Dynamic Execution** (широкое динамическое исполнение).
2. **Технология Intel Advanced Digital Media Boost** (улучшенные цифровые медиа возможности).
3. **Технология Intel Advanced Smart Cache** (улучшенный интеллектуальный кэш).
4. **Технология Intel Smart Memory Access** (интеллектуальный доступ к памяти).
5. **Технология Intel Intelligent Power Capability** (интерактивное подключение подсистем).

Рассмотрим подробнее каждую из перечисленных технологий.

Технология Intel Wide Dynamic Execution

Динамическое исполнение команд подразумевает суперскалярную архитектуру, способную выполнять анализ потока команд и обладающую возможностями спекулятивного (упреждающего) и внеочередного исполнения команд.

В новой архитектуре с «широким» динамическим исполнением связывают, во-первых, возможность исполнения большего числа операций за такт, чем это было раньше. Благодаря добавлению в каждое ядро декодеров и исполнительных устройств каждое из ядер сможет выбирать из программного кода и исполнять до четырех x86 инструкций одновременно с помощью 14-стадийных конвейеров, в то время как предыдущие процессоры Intel, AMD (как настольные, так и мобильные) могли обрабатывать не более трех инструкций за такт. На 4 декодера (один для сложных инструкций и три – для простых) микроархитектура Core предполагает наличие 6 портов запуска (один – Load, два – Store и три – универсальных) исполнительных устройств.

Кроме того, микроархитектура Core получила более совершенный блок предсказания переходов и более вместительные буферы команд, используемые на различных этапах анализа кода для оптимизации скорости исполнения.

Во-вторых, в дополнении к весьма удачной технологии micro-ops fusion (x86 инструкция распадается на последовательность микрокоманд, которые выполняются процессором в этой же последовательности) микроархитектура Core получила технологию macro fusion. Данная технология направлена на увеличение числа исполняемых за такт команд и заключается в том, что ряд пар связанных между собой последовательных x86 инструкций, таких как, например, сравнение со следующим за ним условным переходом, представляются декодером одной микрокомандой. Таким образом, пять выбранных x86 инструкций могут в каждом такте преобразовываться в четыре микрокоманды. Этим достигается как увеличение темпа исполнения кода, так и некоторая экономия энергии.

Технология Intel Advanced Digital Media Boost

До настоящего времени процессоры Intel исполняли одну SSE-инструкцию (SSE, SSE2, SSE3), работающую с 128-битными операндами, лишь за 2 такта. Один такт тратился на обработку старших 64 бит, второй такт – на обработку младших. Новая же микроархитектура позволяет ускорить работу с SSE-инструкциями в два раза. Блоки SSE в данных процессорах полностью 128-битные, что дает возможность увеличить количество данных, обрабатываемых процессором за такт.

Кроме этого, Intel в очередной раз провел ревизию системы команд SSE. Результатом стало расширение SSSE3 еще 32-мя новыми командами, а для процессоров (Penryn), выполненных по 45 нм технологическому процессу, использование нового набора команд SSE4.1, в который

добавлено 47 новых команд, позволяющих ускорить в том числе кодирование видеозаписей с высоким разрешением и обработку фотоизображений.

Технология Intel Advanced Smart Cache

Двухядерные процессоры с микроархитектурой Core имеют разделяемый между двумя ядрами L2 кэш. Плюсов такого подхода несколько:

1. Появляется возможность у процессора гибко регулировать размеры областей кэша, используемых каждым из ядер. Доступ ко всему объему L2 кэша может получить любое из ядер процессора (когда одно из ядер бездействует). Если же одновременно работают два ядра, то кэш делится между ними пропорционально, в зависимости от частоты обращений каждого ядра к оперативной памяти.

Если оба ядра работают синхронно с одними и теми же данными, то хранятся они в общем L2 кэше только однократно. Таким образом, разделяемый интеллектуальный L2 кэш гораздо более эффективен и, даже можно сказать, более вместителен, чем два отдельных кэша, разделенных между ядрами.

2. Значительно снижается нагрузка на оперативную память системы и на процессорную шину. В этом случае перед системой не стоит задача контроля и обеспечения когерентности кэш-памяти различных ядер.

Технология Intel Smart Memory Access

Под этим названием объединены несколько технологий.

1. **Усовершенствованный алгоритм предварительной выборки данных.** Микроархитектура Core предполагает реализацию в процессоре 6 независимых блоков предварительной выборки данных. Два блока нагружаются задачей предварительной выборки данных из памяти в общий L2 кэш. Еще по два блока работают с кэшами L1 каждого ядра. Каждый из этих блоков независимо друг от друга отслеживает закономерные обращения (потоковые либо с постоянным шагом внутри массива) исполнительных устройств к данным. Базируясь на собранной статистике, блоки предварительной выборки стремятся подгружать данные из памяти в процессорный кэш еще до того, как к ним последует обращение. Также L1 кэш каждого из ядер процессоров, построенных на базе Core, имеет по одному блоку предварительной выборки инструкций, работающих по аналогичному алгоритму.

2. **Memory disambiguation (устранение противоречий при доступе к памяти).** Данная технология направлена на повышение эффективности работы алгоритмов внеочередного исполнения инструкций, осуществляющих чтение/выгрузку (Load) и запись/сохранение (Store) данных в память.

Технология Intel Intelligent Power Capability

При разработке новой микроархитектуры Core инженеры стремились к оптимизации параметра «производительность на ватт». Поэтому они сразу предусмотрели набор технологий, направленных на снижение энергопотребления и тепловыделения, в первую очередь хорошо зарекомендовавшие себя технологии **Intel Speed Step** (динамическое изменение тактовой частоты процессора в зависимости от текущих потребностей в вычислительной мощности), **Halt State** (отключает некоторые блоки процессора во время их бездействия) и др.

Но главная особенность новой архитектуры в том, что процессоры получили возможность интерактивного подключения тех собственных подсистем, которые используются в данный момент. Причем речь в данном случае идет не о ядрах целиком. Декомпозиция процессора на отдельные функциональные узлы выполнена на гораздо более низком уровне. Каждое из процессорных ядер поделено на большое количество блоков и внутренних шин, питание которыми управляется раздельно посредством специализированных дополнительных логических схем.

Недостатки микроархитектуры Intel Core

Существенным недостатком процессоров микроархитектуры Intel Core стал их **немодульный дизайн** (**немодульное проектирование**). Они изначально проектировались как двухъядерные полупроводниковые кристаллы. Последующий же переход к выпуску многоядерных представителей Core 2 стал выявлять слабые места такого подхода. Так, 4-ядерные и 6-ядерные представители микроархитектуры Intel Core просто собирались из нескольких 2-ядерных кристаллов, что приводило к затруднению взаимодействия между ними. Обмен данными между разрозненными ядрами организовывался через системную память, что порой вызывало большие задержки, обусловленные ограниченной пропускной способностью процессорной шины.

Еще одно узкое место возникало в многопроцессорных системах. Хотя Intel уже решил проблему с разделением системной шины, выпустив чипсеты, предлагающие собственную шину каждому процессору, **производительность часто ограничивалась недостаточно высокой пропускной способностью шины памяти.**

Таким образом, дальнейшее увеличение многоядерности и многопроцессорности, выбранное основным вектором увеличения производительности современных систем, рано или поздно должны были завести Intel в тупик, даже несмотря на то, что сама по себе микроархитектура Intel Core представляется очень удачной.

3.4.3. Микроархитектура Intel Nehalem

Микроархитектура Nehalem является дальнейшим развитием рассмотренной выше микроархитектуры Intel Core.

Основные отличительные черты Nehalem

Основными отличительными чертами данной микроархитектуры являются следующие:

1. Усовершенствованное по сравнению с Core вычислительное ядро.
2. Многопоточная технология SMT (Simultaneous Multi-Threading), позволяющая исполнять одновременно два вычислительных потока на одном ядре.
3. Три уровня кэш-памяти: L1 кэш размером 64 Кбайта на каждое ядро, L2 кэш размером 256 Кбайт на каждое ядро, общий разделяемый L3 кэш размером 4, 8 и до 24 Мбайт.
4. Интегрированный в процессор контроллер памяти с поддержкой нескольких каналов DDR3 SDRAM.
5. Новая шина QPI с топологией *точка – точка* для связи процессора с чипсетом и процессоров между собой.
6. Модульная структура.
7. Монолитная конструкция – процессор состоит из одного полупроводникового кристалла.
8. Технологический процесс с нормами производства – не более 45 нм.
9. Использование двух, четырех или восьми ядер.
10. Управление питанием и Turbo-режим.

Усовершенствования вычислительного ядра

Несмотря на то, что процессоры семейства Nehalem преподносятся Intel как носители новой микроархитектуры, основная их часть – вычислительное ядро – по сравнению с Core изменилась не столь значительно, наибольшие улучшения кроются в инфраструктуре.

На рис. 3.8 представлена обобщенная структура ядра процессора с микроархитектурой Intel Nehalem.

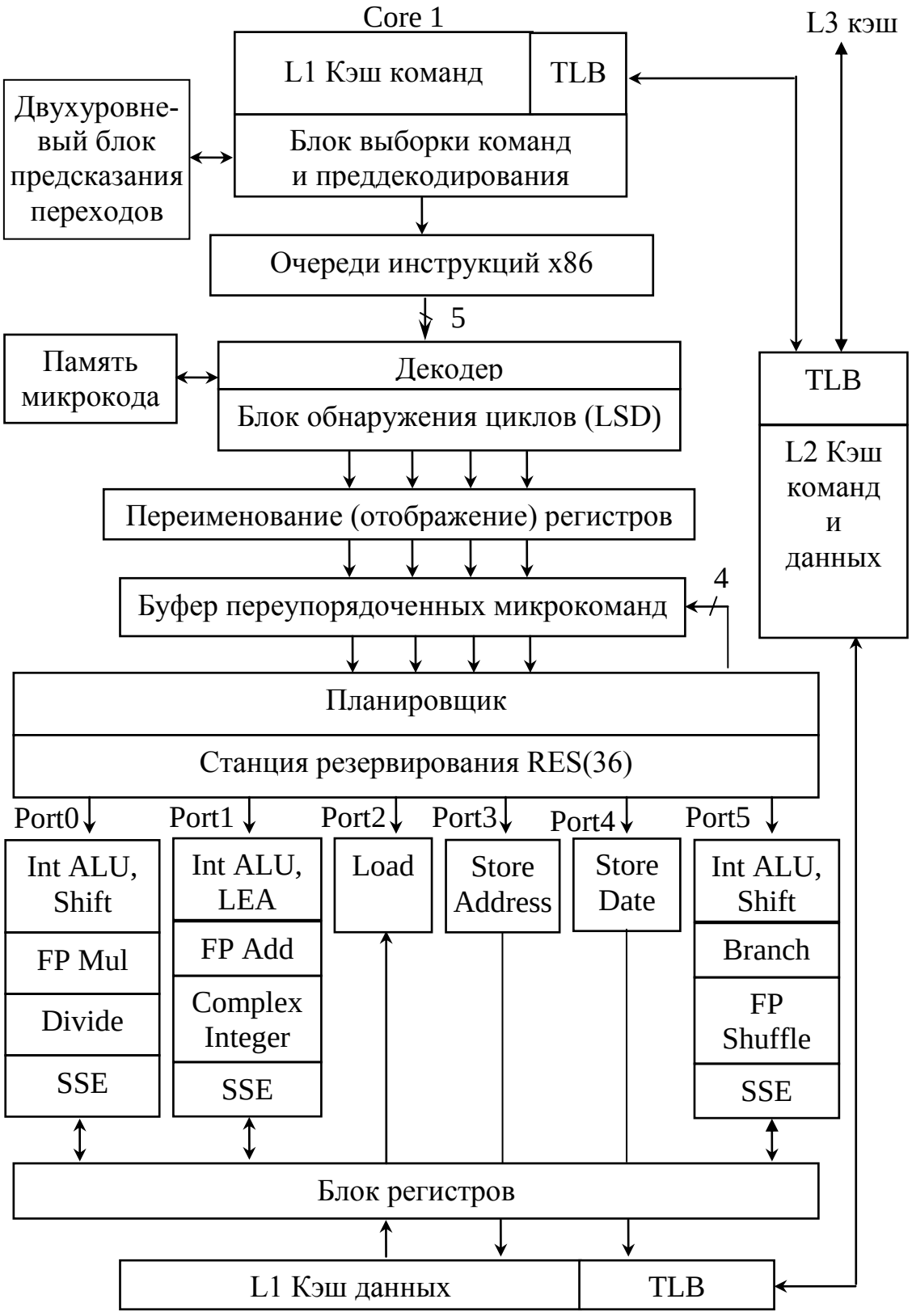


Рис. 3.8. Структура ядра процессора микроархитектуры Nehalem

В рассматриваемом ядре так называемый предпроцессор содержит следующие блоки: блок выборки команд и преддекодирования; блок предсказания переходов (ветвлений); блок очередей инструкций; декодер инструкций; блок обнаружения циклов в программе.

Сначала x86 инструкции выбираются (Fletch) из кэш-памяти команд. Если в потоке команд оказывается команда условного перехода (ветвление программы), то включается механизм предсказания ветвления, который формирует адрес следующей выбираемой команды до того, как будет определено условие выполнения перехода. Если предсказывается выполнение ветвления, то выбирается и загружается в конвейер команда, размещенная по предсказанному адресу.

В дополнение к уже имеющемуся в Intel Core блоку предсказания переходов был добавлен в Nehalem ещё один «предсказатель» второго уровня. Он работает медленнее, чем первый, но зато благодаря более вместительному буферу, накапливающему статистику переходов, обладает лучшей глубиной анализа.

Далее, разделенные x86 инструкции (Pre Decode) на простые и сложные организуются в виде очередей (Instruction Queues) на входах четырех декодеров. Декодеры преобразуют x86 команды в микрокоманды, под управлением которых в процессоре выполняются элементарные операции (микрооперации). Как в Intel Core, три декодера используются для обработки простых инструкций, один – для сложных. Каждая простая x86 инструкция преобразуется в 1–2 микрокоманды, а для сложной инструкции из памяти микрокода (u Code ROM) выбирается последовательность микрокоманд (микропрограмма), которая содержит более двух микрокоманд (технология micro-ops fusion). Используя технологию macro fusion, четыре декодера могут обработать одновременно пять x86 команд, преобразуя их в четыре микрокоманды.

В Nehalem увеличилось число пар x86 команд, декодируемых в рамках этой технологии «одним махом». Кроме того, технология macro fusion стала работать и в 64-битном режиме, в то время как в процессорах семейства Core 2 она могла активироваться лишь при работе процессора с 32-битным кодом.

Следующее усовершенствование, связанное с повышением продуктивности начальной части исполнительного конвейера, коснулось блока обнаружения циклов в программе Loop Stream Detector. Этот блок появился впервые в процессорах с микроархитектурой Core и предназначался для ускорения обработки циклов. Определяя в программе циклы небольшой длины, Loop Stream Detector (LSD) сохранял их в специальном буфере, что давало возможность процессору обходиться без их многократной выборки из кэша и предсказания переходов внутри этих

циклов. В процессорах Nehalem блок LSD стал ещё более эффективен благодаря его переносу за стадию декодирования инструкций. Иными словами, теперь в буфере LSD сохраняются циклы в декодированном виде, из-за чего этот блок стал несколько похож на Trace Cache процессоров Pentium 4. Однако Loop Stream Detector в Nehalem – это особенный кэш. Во-первых, он имеет очень небольшой размер, всего 28 микроопераций, во-вторых, в нём сохраняются исключительно циклы.

После декодирования производится переименование регистров, переупорядочение (Retirement Unit) и сохранение до момента выполнения 128 микрокоманд в буфере (Reorder Buffer). Это количество микрокоманд на 33 % больше, чем в Intel Core (96 микрокоманд).

На следующем этапе планировщик (Scheduler) через станцию резервирования (Reservation Station – RES) вместимостью до 36 инструкций (Intel Core – 32 инструкции) отправляет микрокоманды непосредственно на исполнительные устройства.

Также как Core 2, процессоры с микроархитектурой Nehalem способны отправлять на выполнение до шести микроопераций одновременно. В каждом ядре процессора Intel Nehalem используются три универсальных порта (Port0, Port1, Port5) для связи с различными исполнительными устройствами, два порта (Port3, Port4) – для организации записи/загрузки (Store) адреса и данных в память и один (Port2) для организации чтения/выгрузки (Load) данных из памяти. Универсальные порты осуществляют связь с тремя блоками – для обработки целочисленных 64-битных данных (ALU), выполнения сдвигов (Shift) и операций сравнения (LEA); с тремя блоками – для обработки чисел с плавающей точкой (FAdd, FMul, FPShuffes); с тремя 128-битными блоками для обработки потоковых данных (SSE); с одним блоком – для исполнения переходов (Branch); со специальными блоками Divide (деление), Complex Integer (сложные целочисленные операции).

В данном процессоре (ядре), как и в любом другом современном процессоре, реализована конвейерная технология обработки команд. Длина каждого из четырех конвейеров составляет 14 ступеней.

В микроархитектуре Nehalem Intel продолжила взятый ранее курс на увеличение числа поддерживаемых SIMD инструкций. Пополненный набор команд расширился за счет семи новых инструкций и получил название SSE4.2. В SSE4.2 добавлено пять инструкций, предназначенных для ускорения синтаксического анализа XML-файлов. Также с помощью этих же инструкций возможно увеличение скорости обработки строк и текстов. Ещё две новые инструкции из набора SSE4.2 нацелены на совершенно иные приложения. Первая из них аккумулирует контрольную сумму, а вторая подсчитывает число ненулевых бит в источнике.

Новая структура кэш-памяти

От двухуровневой структуры кэш-памяти в Intel Core с общим на каждые два ядра L2 кэшем в процессорах Nehalem остался только кэш первого уровня суммарным объёмом 64 Кб, который делится на две равные части для хранения инструкций и данных. Использование разделяемого L2 кэша оказалось весьма проблематичным при увеличении количества ядер, и поэтому в микроархитектуре Nehalem, предполагающей наличие в процессоре до 8 ядер, кэш второго уровня не является разделяемым. Каждое из ядер получило свой собственный L2 кэш со сравнительно небольшим объёмом – 256 Кбайт.

К двум уровням кэша в Nehalem добавился и L3 кэш, который объединяет ядра между собой и является разделяемым. В результате L2 кэш выступает буфером при обращениях процессорных ядер в разделяемую кэш-память, имеющую достаточно большой объём.

Кэш L3 в Nehalem работает на более высокой частоте, которая для первых представителей этого семейства установлена равной 2,66 ГГц. Разработчики Intel не стал отказываться от дублирования данных, хранящихся в кэшах первого и второго уровней, в L3 кэше, что обеспечивает в многоядерных процессорах более высокую скорость работы подсистемы памяти.

Несмотря на кардинальный пересмотр системы кэширования, алгоритмы работы блоков предварительной выборки не изменились, они в Nehalem целиком позаимствованы из Intel Core. Это означает, что упреждающая выборка данных и инструкций производится только в кэш-память первого и второго уровня. Даже при использовании старых алгоритмов, результативность работы блоков предварительной выборки улучшилась. Объясняется это тем, что L2 кэш в Nehalem индивидуален для каждого ядра, а при такой организации кэш-памяти гораздо легче отслеживать шаблоны в обращениях. Благодаря появлению L3 кэша работа блока предварительной выборки не наносит существенного ущерба пропускной способности шины памяти.

Кроме того, существенно увеличился размер TLB (Translation-Lookaside Buffer). TLB – это высокоскоростной буфер, который используется для установления соответствия между виртуальными и физическими адресами страниц. Увеличение размера TLB, таким образом, позволяет повысить число страниц памяти, которые могут быть одновременно использованы без дополнительных дорогостоящих преобразований по таблицам трансляции адресов, находящимся в обычной памяти. Более того, TLB в процессорах Nehalem стал двухуровневым. Фактически к унаследованному от процессоров Core 2 TLB был добавлен ещё

один буфер второго уровня. При этом новый L2 TLB отличается не только высокой вместительностью, позволяющей сохранять до 512 записей, но и сравнительно низкой латентностью. Ещё одна особенность L2 TLB заключается в том, что он унифицирован и способен транслировать адреса страниц любого размера. Изменения в системе TLB сделаны в первую очередь с прицелом на серверные приложения, активно оперирующие большими объёмами памяти. Однако и в «настольных» задачах увеличенное число вхождений TLB может оказать положительное влияние на быстродействие подсистемы памяти.

Реализация многопоточности

Возвращение в Nehalem технологии SMT – одно из самых существенных нововведений, способных положительно повлиять на производительность (в процессорах Pentium 4 эта же технология преподносилась под маркетинговым именем Hyper-Threading).

Внедрение SMT в Nehalem не потребовало существенного увеличения сложности процессора. Продублированы в ядре фактически лишь процессорные регистры. Все остальные ресурсы при включении SMT разделяются в процессоре между потоками динамически (например, Reservation Station или кэш-память) либо жёстко пополам (например, Reorder Buffer). Как и в процессорах Pentium 4, активация SMT в Nehalem приводит к тому, что каждое физическое ядро видится операционной системой как пара логических ядер. Например, четырёхъядерный Nehalem будет распознаваться программным обеспечением как процессор с восемью ядрами.

Интегрированный в процессор контроллер памяти

Nehalem стала первой интеловской микроархитектурой, предполагающей интеграцию контроллера памяти внутрь процессора. Главное свойство контроллера памяти процессоров семейства Nehalem – гибкость. Учитывая модульный дизайн всего семейства процессоров, которое может содержать сильно различающиеся по характеристикам и рыночному позиционированию продукты, Intel предусмотрела возможность не только включать или отключать поддержку буферизированных модулей, но и варьировать число каналов и скорость памяти. При этом первые процессоры с микроархитектурой Nehalem в четырёхъядерном варианте получили трёхканальный контроллер памяти с поддержкой DDR3 SDRAM. Пропускная способность подсистемы памяти, в случае использования трёх модулей DDR3, достигает 25,6 Гбайт/с.

Основное преимущество переноса контроллера DRAM в процессор заключается не столько в росте пропускной способности, сколько в уменьшении латентности подсистемы памяти. Ещё одно косвенное преимущество встроенного в процессор контроллера памяти заключается в том, что его функционирование теперь не зависит ни от чипсета, ни от материнской платы. В результате Nehalem показывает одинаковую скорость работы с памятью при работе в платформах различных разработчиков и производителей.

Новая процессорная шина QPI

Микроархитектура Nehalem универсальна, она используется как в настольных, мобильных, так и в серверных продуктах. Поэтому при разработке данной микроархитектуры было уделено внимание проектированию новой процессорной шины, которая оказалась бы применима в многопроцессорных системах, обеспечивая необходимую пропускную способность и масштабируемость. Используемая ранее шина FSB в многопроцессорных системах оказывается неприменима, необходимо использовать «распределенную» модель памяти NUMA (Non-Uniform Memory Access), а следовательно, нужно прямое и высокоскоростное соединение между процессорами.

Для решения этой задачи был построен специальный последовательный интерфейс CSI (Common System Interface) с топологией точка-точка, переименованный впоследствии в QPI (QuickPath Interconnect). С технической точки зрения шина QPI представляет собой два 20-битных соединения, ориентированных на передачу данных в прямом и обратном направлении; 16 бит предназначаются для передачи данных, оставшиеся четыре носят вспомогательный характер, они используются протоколом и коррекцией ошибок. Эта шина работает на максимальной скорости 6,4 миллиона передач данных в секунду (GT/s) и имеет, соответственно, пропускную способность 12,8 Гбайт/с в каждую сторону, или 25,6 Гбайт/с суммарно. В зависимости от рыночного ориентирования процессоры с микроархитектурой Nehalem могут комплектоваться одним или несколькими интерфейсами QPI. В итоге в многопроцессорной системе каждый из процессоров может иметь прямую связь с конечным числом процессоров системы.

Модульная структура процессора

Важным нововведением в Nehalem стал модульный дизайн процессора. Фактически микроархитектура сама по себе включает лишь не-

сколько «строительных блоков», из которых на этапе конечного проектирования и производства может быть собран итоговый процессор. Этот набор строительных блоков включает в себя (рис. 3.9) процессорное ядро с L2 кэшем (Core), L3 кэш, контроллер шины (QPIC), контроллер памяти (MC), графическое ядро (GPU), контроллер потребляемой энергии (PCU) и т.д.

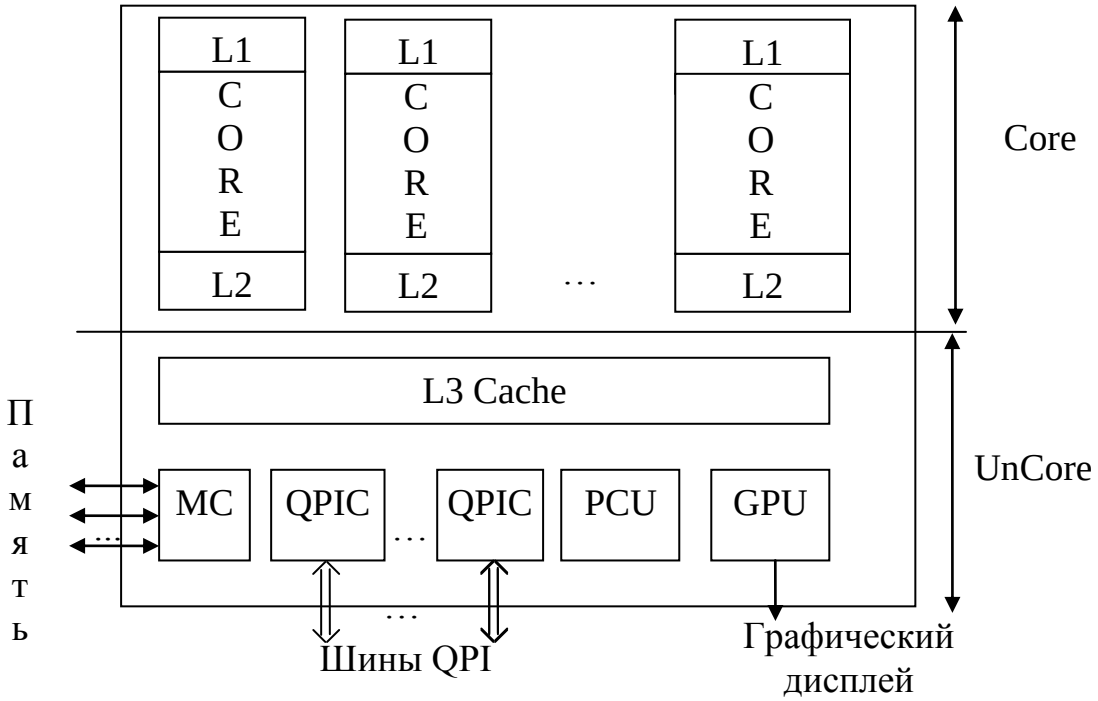


Рис. 3.9. Модульная структура процессора

Необходимые «кубики» собираются в едином полупроводниковом кристалле и преподносятся в качестве решения для того или иного рыночного сегмента.

Первыми серийными процессорами, основанными на новой микроархитектуре Nehalem, стали настольные модели, известные под кодовым именем Bloomfield. Эти процессоры имеют четырёхядерное строение. Помимо процессорных ядер, в полупроводниковый кристалл Bloomfield включен кэш третьего уровня объёмом 8 Мбайт, трёхканальный контроллер памяти с поддержкой DDR3 SDRAM и один интерфейс QPI. Процессоры с ядром Bloomfiled продаются под именем Core i7 серия 9xx.

Следующая линейка процессоров Intel Core i7-8xx с ядром Lynnfield отличается от предыдущей двухканальным интегрированным

контроллером памяти, уменьшенным TDP, увеличенной частотой шины памяти 1066/1333 МГц, встроенным в процессор контроллером шины графического адаптера PCI Express x16, использованием системной шины DMI для связи процессора с южным мостом чипсета, более агрессивной реализацией технологии Turbo Boost.

Управление питанием и Turbo-режим

Многие изменения, реализованные инженерами Intel в процессорах Nehalem, связаны с оптимизацией микроархитектуры под врожденное многоядерное строение. Поэтому необходимость пересмотра системы управления питанием процессора назрела сама собой. Многоядерные процессоры с микроархитектурой Core очень неэкономичны с той точки зрения, что управление энергосбережением в них происходит по единому алгоритму, который практически не учитывает состояния отдельных ядер. И поэтому, например, нередки ситуации, когда одно находящееся под вычислительной нагрузкой ядро препятствует переходу в энергосберегающие состояния остальных ядер, несмотря на то, что они фактически простаивают.

Именно поэтому микроархитектура Nehalem предполагает наличие в процессоре еще одного важного блока – PCU (Power Control Unit). Этот блок представляет собой встроенный в процессор программируемый микроконтроллер (т.е., по сути, процессор в процессоре), целью которого является «интеллектуальное» управление потреблением энергии. Неудивительно, что при этом PCU имеет достаточно сложную конструкцию: на его реализацию ушел примерно 1 миллион транзисторов.

Основным предназначением PCU является **управление частотой и напряжением питания отдельных ядер**, для чего этот блок имеет все необходимые средства. Он получает от всех ядер со встроенных в них датчиков всю информацию о температуре, напряжении и силе тока.

Основываясь на этих данных, PCU может переводить отдельные ядра в энергосберегающие состояния, а также управлять их частотой и напряжением питания. В частности, PCU может независимо друг от друга отключать неактивные ядра, переводя их в состояние глубокого сна, в котором энергопотребление ядра приближается к нулевой отметке.

Главное преимущество этой технологии состоит в том, что управление питанием отдельных ядер осуществляется целиком внутри процессора и не требует усложнения схемы конвертера питания на материнской плате.

Что же касается общих для всех ядер процессорных блоков, таких как контроллеры памяти и интерфейса QPI, то они переходят в энерго-

сберегающие состояния, когда в состоянии сна находятся все процессорные ядра.

Технология Turbo Boost

Наличие в процессоре контроллера, способного независимо управлять состоянием процессорных ядер, позволило Intel реализовать и еще одну интересную технологию, получившую название Turbo Boost Technology. Эта технология вводит понятие турбо-режима, в котором отдельные ядра могут работать на частоте, превосходящей номинальную, т.е. разгоняться. Основной принцип Turbo Boost Technology состоит в том, что при переходе отдельных ядер в энергосберегающие состояния снижается общее энергопотребление и тепловыделение процессора, а это, в свою очередь, позволяет нарастить частоты остальных ядер без риска выйти за установленные рамки TDP.

Фактически прообраз этой технологии уже был реализован в двухъядерных мобильных процессорах поколения Penryn, однако в Nehalem её развитие продвинулось еще дальше. В новых процессорах, если нет риска выйти за границу типичного энергопотребления и тепловыделения, PCU может повышать частоты процессорных ядер на один шаг выше номинала (133 МГц). Это может происходить, например, при слабо распараллеленной нагрузке, когда часть ядер находится в состоянии простоя. Более того, при соблюдении описанных условий, частота одного из ядер может быть увеличена и на два шага выше номинала (266 МГц).

Следует отметить, что необходимым условием включения турборежима вовсе не является переход одного или нескольких ядер в энергосберегающее состояние. Это всего лишь один из возможных сценариев. Так как PCU имеет все средства для получения данных о фактическом состоянии процессорных ядер, турбо-режим может задействоваться и в тех случаях, когда все ядра работают, но нагрузка на часть из них невелика.

Большим преимуществом Turbo Boost Technology является ее полная прозрачность для операционной системы. Эта технология реализована исключительно аппаратными средствами и не требует использования никаких программных утилит для своей активации.

3.4.4. Семейство процессоров Intel Westmere

Основываясь на ошеломляющем успехе 45 нм производственного процесса с диэлектриками high-k и транзисторами с металлическими за-

творами, корпорация Intel в конце 2009 г. запустила 32 нм производственную технологию, в которой используются диэлектрики high-k и транзисторы с металлическими затворами **второго поколения**. Эта технология стала основой для новой 32 нм версии микроархитектуры Intel Nehalem. Новые процессоры Intel семейства Westmere стали первыми процессорами, созданными по нормам 32 нм техпроцесса. Эти процессоры известны под кодовыми названиями Clarkdale и Arrandale, предназначены для применения соответственно в настольных компьютерах и ноутбуках и входят в модельные линейки Intel Core i3, i5, i7. Процессоры Intel Westmere представляют собой двухъядерные решения. Кроме того, в их конструкции присутствуют два несущих кристалла (рис. 3.10), один из которых, выпускаемый по 32 нм техпроцессу, включает в себя два вычислительных ядра, разделяемую L3 кэш-память, контроллер шины QPI.

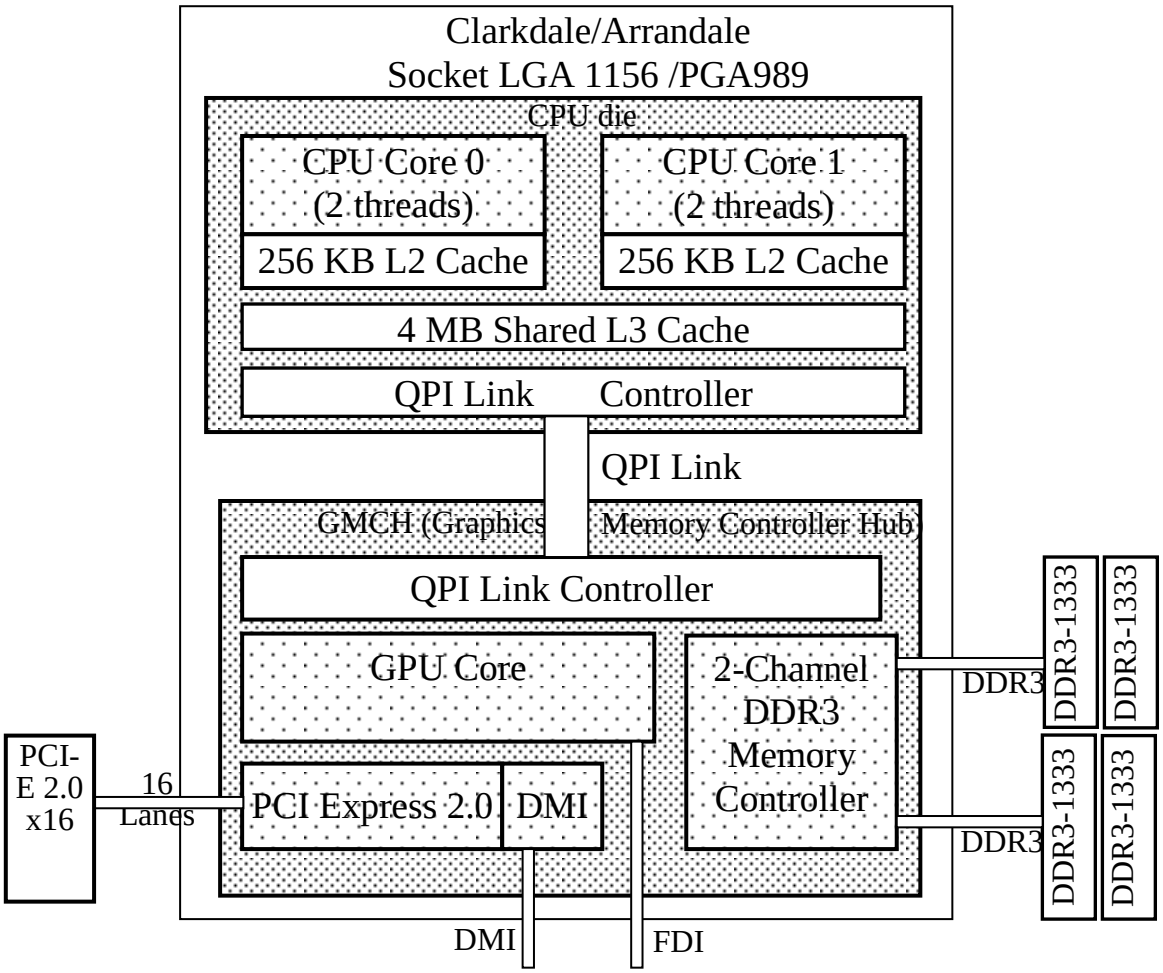


Рис. 3.10. Структура процессоров Intel семейства Westmere

Второй, более крупный кристалл, изготавливаемый по 45 нм технологии, содержит графический процессор GPU, двухканальный контроллер памяти DDR3, контроллер интерфейса PCI Express 2.0 и контроллер шин DMI и FDI (Flexible Display Interface). Взаимодействие между двумя кристаллами происходит по высокоскоростной шине QPI.

3.4.5. Микроархитектура Sandy Bridge

Ключевыми особенностями процессоров архитектуры Sandy Bridge по сравнению с Nehalem являются:

- Усовершенствованное вычислительное ядро (рис. 3.11).

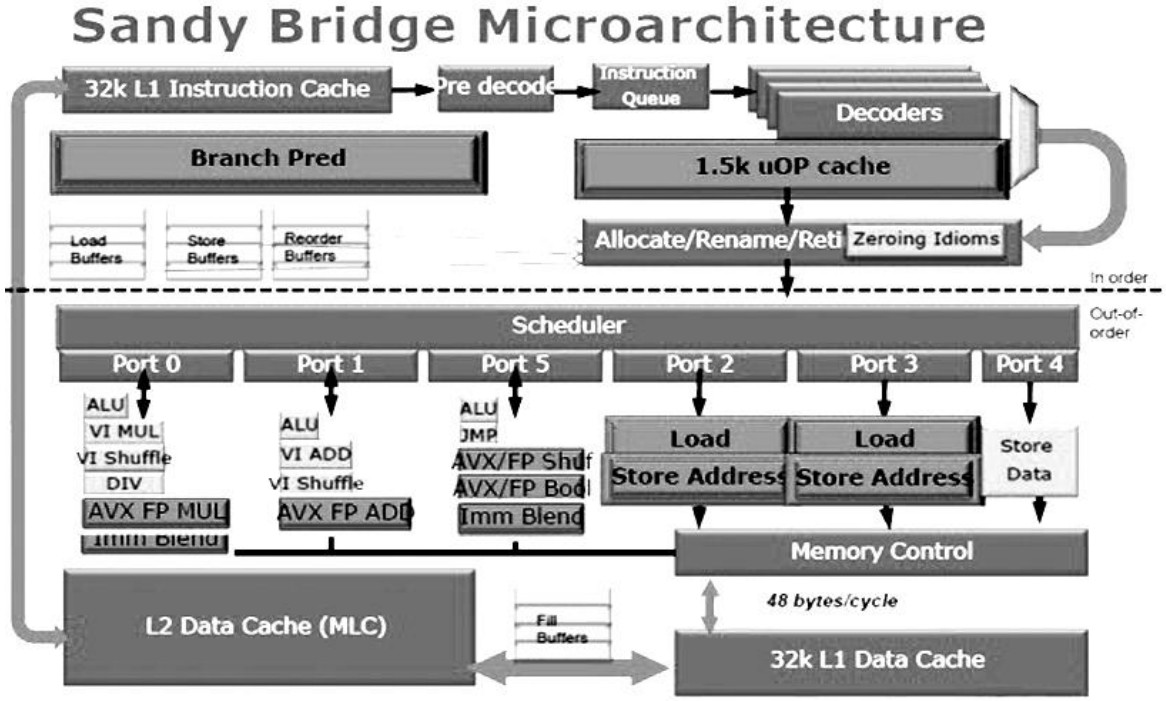


Рис. 3.11. Структура ядра микроархитектуры Sandy Bridge

- Монолитная конструкция – процессор состоит из одного полупроводникового кристалла, изготовленного по 32 нм технологии техпроцесса.
- Новый набор инструкций Intel Advanced Vector Extensions (AVX) для ускорения обработки вещественных чисел.
- Оптимизированная технология Intel Turbo Boost.
- Заметно увеличившаяся энергоэффективность.
- Производительность интегрированного в процессор графического ядра значительно увеличена.

- Новая кольцевая шина Ring Interconnect.
- Наличие нового функционального узла процессора – системного агента.
- Усовершенствованный интегрированный контроллер памяти.

Более подробно особенности организации процессорного ядра Sandy Bridge рассмотрены в разделе 3.4.6. Рассмотрение проводится в сравении с ядром новой микроархитектуры Intel Haswell.

Кольцевая шина

Вся история модернизации процессорных микроархитектур Intel последних лет неразрывно связана с последовательной интеграцией в единый кристалл всё большего количества модулей и функций, ранее располагавшихся вне процессора: в чипсете, на материнской плате и т.д. Соответственно, по мере увеличения производительности процессора и степени интеграции чипа, требования к пропускной способности внутренних межкомпонентных шин росли опережающими темпами.

В микроархитектуре Sandy Bridge для повышения общей производительности системы разработчики решили обратиться к кольцевой топологии 256-битной межкомпонентной шины, выполненной на основе новой версии технологии QPI (QuickPath Interconnect), расширенной и доработанной (рис. 3.12).

Кольцевая шина (Ring Interconnect) в версии архитектуры Sandy Bridge для настольных и мобильных систем служит для обмена данными между шестью ключевыми компонентами чипа: четырьмя процессорными ядрами x86, графическим ядром, кэш-памятью L3, теперь её называют LLC (Last Level Cache – кеш последнего уровня), и системным агентом. Шина состоит из четырёх 32-байтных колец: шины данных (Data Ring), шины запросов (Request Ring), шины мониторинга состояния (Snoop Ring) и шины подтверждения (Acknowledge Ring).

Управление шинами осуществляется с помощью коммуникационного протокола распределённого арбитража, при этом конвейерная обработка запросов происходит на тактовой частоте процессорных ядер, что придаёт архитектуре дополнительную гибкость при разгоне. Производительность кольцевой шины оценивается на уровне 96 Гбайт в секунду на соединение при тактовой частоте 3 ГГц, что фактически в четыре раза превышает показатели процессоров Intel предыдущего поколения.

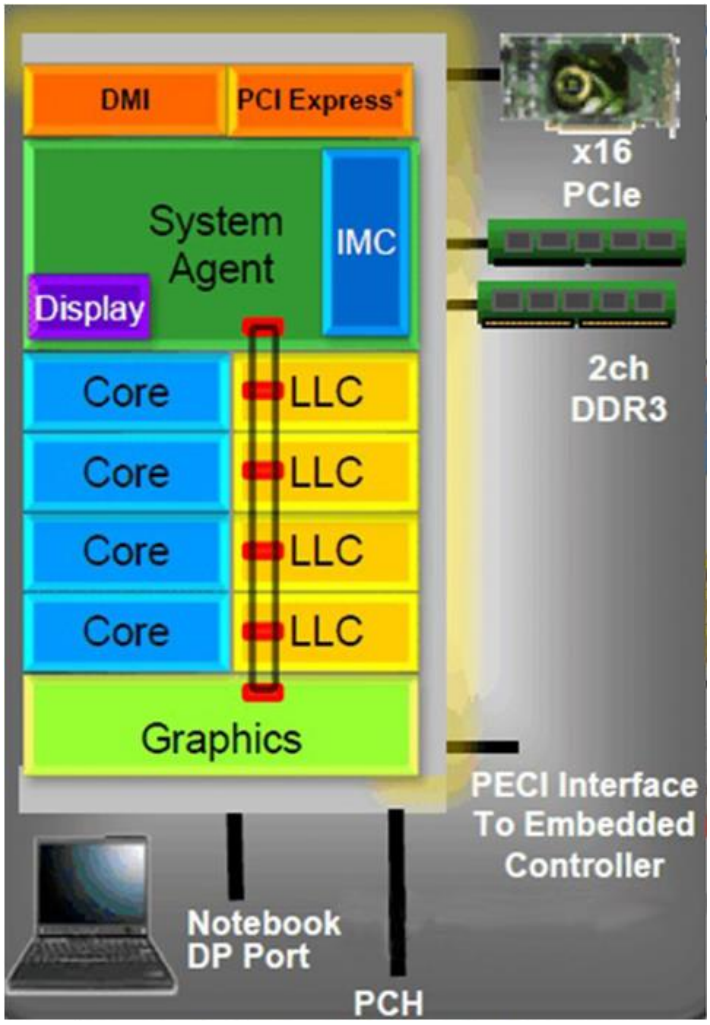


Рис. 3.12. Кольцевая шина SandyBridge

Кольцевая топология и организация шин обеспечивает минимальную латентность при обработке запросов, максимальную производительность и отличную масштабируемость технологии для версий чипов с различным количеством ядер и других компонентов. По словам представителей компании, в перспективе к кольцевой шине может быть "подключено" до 20 процессорных ядер на кристалл, и подобный редизайн, как вы понимаете, может производиться очень быстро, в виде гибкой и оперативной реакции на текущие потребности рынка. Кроме того, физически кольцевая шина располагается непосредственно над блоками кеш-памяти L3 в верхнем уровне металлизации, что упрощает разводку дизайна и позволяет сделать чип более компактным.

3.4.6. Микроархитектура Intel Haswell

Вычислительное ядро Haswell не претерпело кардинальных изменений в сравнении с вычислительным ядром Sandy Bridge — были улучшены лишь отдельные блоки ядра процессора. А потому уместным будет рассмотреть в общих чертах микроархитектуру Sandy Bridge и остановиться на внесенных в нее изменениях в Haswell.

Традиционно описание микроархитектуры ядра процессора начинается с блока предпроцессора (front-end), который отвечает за выборку инструкций x86 из кэша инструкций и их декодирование. В микроархитектуре Haswell блок предпроцессора претерпел минимальные изменения (рис. 3.13).

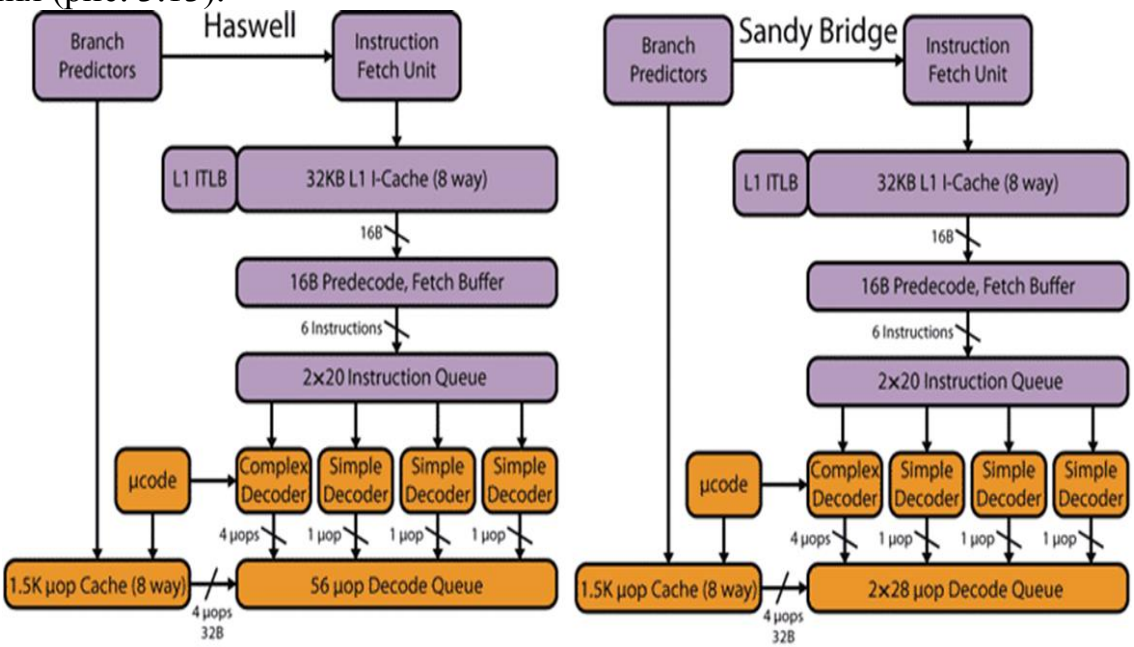


Рис. 3.13. Структура предпроцессора

Инструкции x86 выбираются из кэша инструкций L1I (Instruction Cache), который не изменился в микроархитектуре Haswell. Он имеет размер 32 Кбайт, является 8-канальным и динамически разделяем между двумя потоками инструкций (поддержка технологии Hyper-Threading).

Из кэша L1I команды загружаются 16-байтными блоками в 16-байтный буфер предекодирования (Fetch Buffer).

Поскольку инструкции x86 имеют переменную длину (от 1 до 16 байт), а длина блоков, которыми команды загружаются из кэша, фиксированная, при декодировании команд определяются границы между отдельными командами (информация о размерах команд хранится в кэше

инструкций L1I в специальных полях). Процедура выделения команд из выбранного блока называется предварительным декодированием (PreDecode).

После операции выборки команды организуются в очередь (Instruction Queue). В микроархитектуре Sandy Bridge и Haswell буфер очереди команд рассчитан на 20 команд в каждом из двух потоков, причем из буфера предекодирования за каждый такт в буфер очереди команд могут загружаться до шести выделенных команд.

После этого выделенные команды (x86-инструкции) передаются в декодер, где они преобразуются в машинные микрокоманды (обозначаются как micro-ops или uOps).

Декодер ядра процессора Haswell остался без изменений. Он попрежнему является четырехканальным и может декодировать в каждом такте до четырех инструкций x86.

Четырехканальный декодер состоит из трех простых декодеров, декодирующих простые инструкции в одну микрокоманду, и одного сложного, который способен декодировать одну инструкцию не более чем в четыре микрокоманды (декодер типа 4-1-1-1). Для еще более сложных инструкций, декодирующихся более чем в четыре микрокоманды, сложный декодер соединен с блоком uCode Sequenser, который и применяется для декодирования подобных инструкций.

При декодировании инструкций используются технологии Macro-Fusion и Micro-Fusion.

Кроме того, в микроархитектуре Haswell и Sandy Bridge применяется кэш декодированных микрокоманд (Uop Cache), в который поступают все декодированные микрокоманды. Этот кэш рассчитан приблизительно на 1500 микрокоманд средней длины.

Концепция кэша декодированных микрокоманд заключается в том, чтобы сохранять в нем уже декодированные последовательности микрокоманд. В результате, если нужно выполнить некую x86-инструкцию повторно, а соответствующая ей последовательность декодированных микрокоманд все еще находится в кэше декодированных микрокоманд, не требуется вторично выбирать эту инструкцию из кэша L1 и декодировать ее — из кэша на дальнейшую обработку поступают уже декодированные микрокоманды.

После процесса декодирования x86-инструкций они, по четыре штуки за такт, поступают в буфер очереди декодированных инструкций (Decode Queue). В микроархитектуре Sandy Bridge этот буфер очереди декодированных инструкций был рассчитан на два потока команд по 28 микрокоманд на каждый поток. В микроархитектурах Ivy Bridge и

Haswell он не делится на два потока команд и рассчитан на 56 микрокоманд. Такой подход оказывается более предпочтительным при выполнении однопоточного приложения (с одним потоком команд). В этом случае одному потоку команд доступен буфер емкостью на 56 микрокоманд, а в микроархитектуре Sandy Bridge — только на 28 микрокоманд.

Предпроцессоры ядер Haswell и Sandy Bridge различаются лишь структурой буфера очереди декодированных инструкций.

Тем не менее, как заявляет компания Intel, некоторые улучшения в предпроцессор Haswell все же были внесены и касались усовершенствования блока предсказания ветвлений (Branch Predictors). Однако, какие именно улучшения были реализованы, компания Intel не раскрывает.

Заканчивая описание предпроцессора в микроархитектуре Haswell, нужно также упомянуть и о TLB-буфере.

Буфер TLB (Translation Look-aside Buffers) — это специальный кэш процессора, в котором сохраняются адреса декодированных инструкций и данных, что позволяет значительно сократить время доступа к ним. Этот кэш предназначен для сокращения времени преобразования виртуального адреса данных или инструкций в физический. Дело в том, что процессор использует виртуальную адресацию, а для доступа к данным в кэше или оперативной памяти нужны реальные физические адреса. Преобразование виртуального адреса в физический занимает приблизительно три такта процессора. TLB-кэш хранит результаты предыдущих преобразований, благодаря чему преобразование адреса возможно осуществлять за один такт.

В процессорах с микроархитектурой Haswell и Sandy Bridge (как и в процессорах Intel на базе других микроархитектур) используется двухуровневый кэш TLB, причем если кэш L2 TLB является унифицированным, то L1 TLB-кэш разделен на буфер данных (DTLB) и буфер инструкций (ITLB).

Блок внеочередного исполнения команд

После процесса декодирования x86-инструкций начинается этап их внеочередного исполнения (Out-of-Order).

На первом этапе происходит переименование и распределение дополнительных регистров процессора, которые не определены архитектурой набора команд. Поэтому из буфера очереди декодированных инструкций (Decode Queue) микрооперации по четыре штуки за такт поступают в буфер переупорядочения (ReOrder Buffer), где происходит

переупорядочение микроопераций не в порядке их поступления (Out-of-Order).

В микроархитектуре Sandy Bridge размер буфера переупорядочения рассчитан на 168 микроопераций, а в микроархитектуре Haswell — на 192 микрооперации (рис. 3.14).

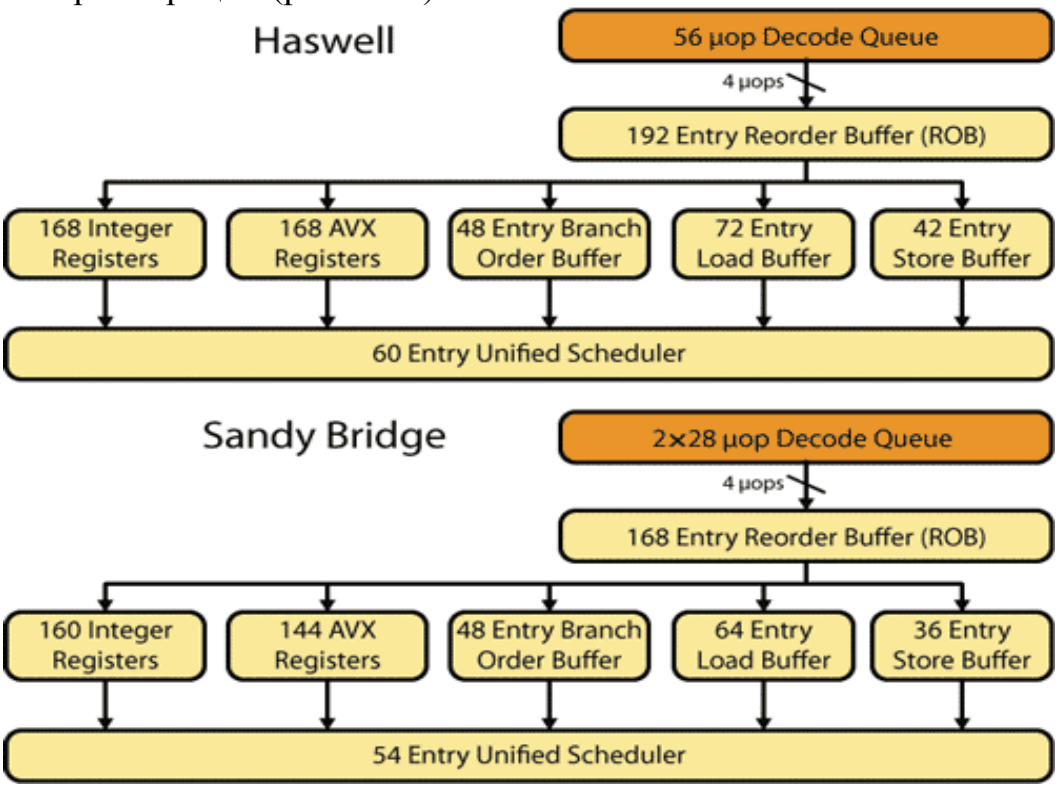


Рис. 3.14. Структура блока внеочередного исполнения команд

Далее происходит распределение микрокоманд по исполнительным блокам. В блоке процессора Unified Scheduler формируются очереди микрокоманд, в результате чего микрокоманды попадают на один из портов функциональных устройств (Dispatch ports). Этот процесс называется диспетчеризацией (Dispatch), а сами порты выполняют функцию шлюза к функциональным устройствам.

В микроархитектурах Sandy Bridge и Haswell кластеры внеочередного выполнения команд (Out-of-Order Cluster) используют так называемые физические регистровые файлы (Physical Register File, PRF), в которых хранятся операнды микроопераций.

Напомним, что, когда в ядрах процессоров не применялись физические регистровые файлы (например, в микроархитектуре Nehalem), каждая микрооперация имела копию необходимого ей операнда (или операндов). Фактически это означало, что блоки кластера внеочередного выполнения команд должны были обладать достаточно большим

размером, чтобы иметь возможность вмещать микрооперации вместе с требуемыми им операндами.

Использование PRF позволяет самим микрооперациям сохранять лишь указатели на операнды, но не сами операнды. С одной стороны, такой подход обеспечивает снижение энергопотребления процессора, поскольку перемещение по конвейеру микроопераций вместе с их операндами требует существенных затрат по энергопотреблению. С другой — применение физического регистрового файла позволяет сэкономить размер кристалла, а высвободившееся пространство использовать для увеличения размеров буферов кластера внеочередного выполнения команд.

В микроархитектуре Sandy Bridge физический регистровый файл для целочисленных операндов (Integer Registers) рассчитан на 160 записей, а для операндов с плавающей запятой (AVX Registers) — на 144 записи.

В микроархитектуре Haswell физические регистровые файлы Integer Registers и AVX Registers рассчитаны на 168 записей.

Буферы чтения (Load) и записи (Store), которые используются для доступа к памяти, также увеличились. Например, если в микроархитектуре Sandy Bridge буферы Load и Store были рассчитаны на 64 и 36 записей соответственно, то в микроархитектуре Haswell они рассчитаны соответственно на 72 и 42 записи.

Размер буфера Unified Scheduler, в котором формируются очереди микроопераций к портам функциональных устройств, также изменился в микроархитектуре Haswell. Если в Sandy Bridge он был рассчитан на 54 микрооперации, то в Haswell — на 60.

Итак, если сравнивать архитектуры Haswell и Sandy Bridge, то в блоке внеочередного исполнения команд микроархитектура Haswell имеет не структурные, а лишь качественные изменения, касающиеся увеличения размеров буферов. Но никаких принципиальных изменений в блоке внеочередного исполнения команд в микроархитектуре Haswell нет.

Исполнительные блоки ядра процессора

Что касается исполнительных блоков ядра процессора, то в микроархитектуре Haswell они претерпели существенные изменения по сравнению с микроархитектурой Sandy Bridge. Так, в Sandy Bridge насчитывается шесть портов функциональных устройств (портов диспетчеризации): три вычислительных и три для работы с памятью.

В микроархитектуре Haswell количество портов функциональных устройств увеличено до восьми.

На рис. 3.15 показаны только вычислительные порты.

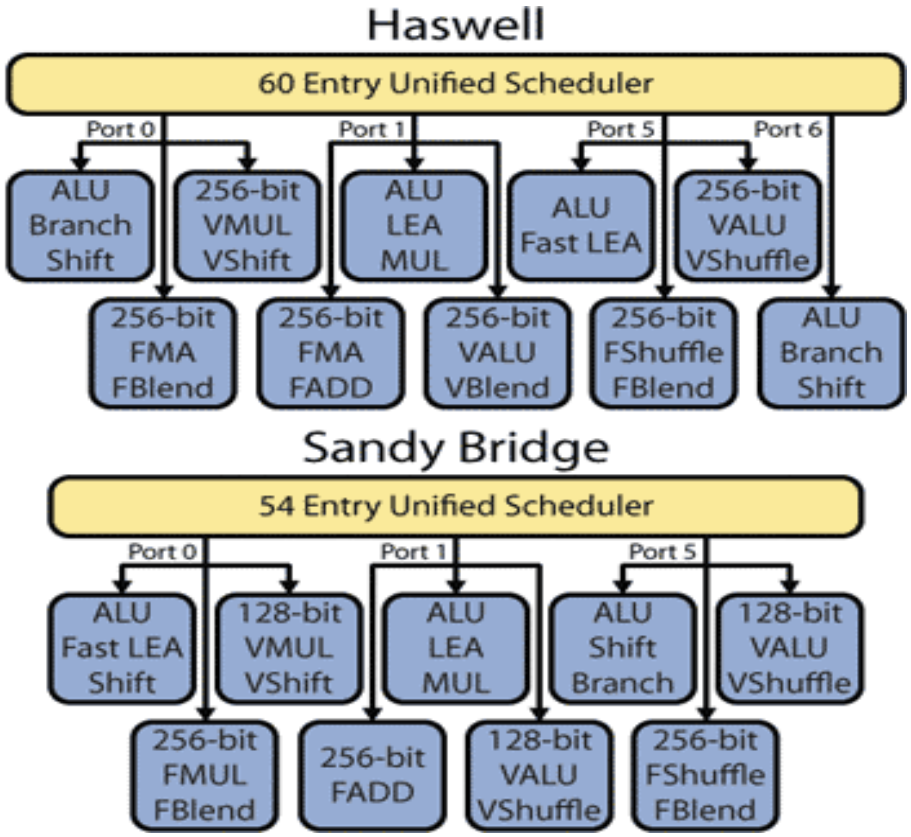


Рис. 3.15. Состав исполнительных утройств, подсоединенных к вычислительным портам

К тому, что было в микроархитектуре Sandy Bridge, добавили еще один порт для записи адреса (Store address) и вычислительный порт для операций с целыми числами и операций сдвига (Integer ALU & Shift). Таким образом, процессоры Haswell могут за один такт выполнять до восьми микроопераций, в то время как в микроархитектуре Sandy Bridge максимальное количество выполняемых за такт микроопераций равно шести.

Кроме того, в микроархитектуре Haswell немного изменены и сами исполнительные устройства. Связано это с тем, что в микроархитектуре Haswell появились дополнительные наборы инструкций: AVX2, FMA3 и BMI.

Подсистема памяти в микроархитектуре Haswell

Одно из наиболее значимых изменений в микроархитектуре Haswell в сравнении с Sandy Bridge было сделано в подсистеме памяти (рис. 3.16). И дело не только в том, что увеличен размер буферов чтения (Load) и записи (Store), которые используются для доступа к памяти (72 и 42 записи соответственно). Главное, был добавлен еще один порт для записи адреса (Store address), кэш данных L1 стал более производительным, а пропускная способность между кэшами L1 и L2 увеличена. Рассмотрим эти изменения более подробно.

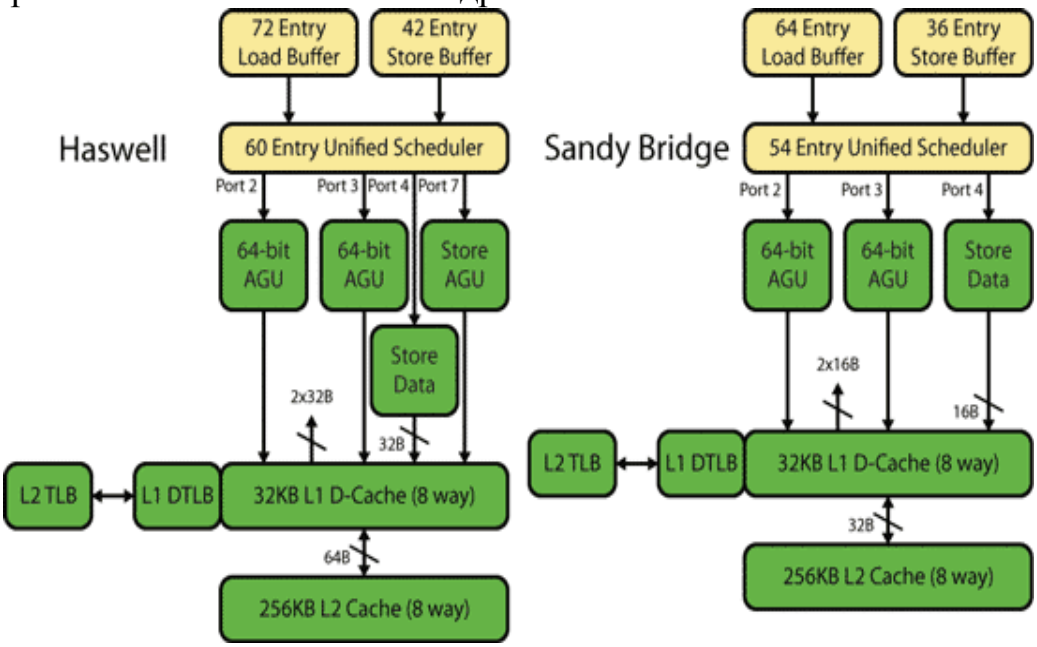


Рис. 3.16 Структура подсистема памяти

Доступ к подсистеме памяти начинается с того, что соответствующие микрооперации поступают в буферы чтения (Load) и записи (Store), которые в совокупности могут накапливать более ста микроопераций. В микроархитектуре Sandy Bridge порты функциональных устройств, которые маркируются на схемах как 2, 3 и 4, отвечали именно за доступ к памяти. Порты 2 и 3 связаны с функциональными устройствами генерации адреса (Address Generation Unit, AGU) для записи или чтения данных, а порт 4 связан с функциональным устройством для записи данных из ядра процессора в кэш данных L1 (DL1). Процедура генерации адреса занимает один или два такта процессора.

В микроархитектуре Haswell к портам 4, 2 и 3 добавлен еще порт 7, который связан с функциональным устройством генерации адреса для записи данных (Store AGU). В результате ядро Haswell может поддер-

живать две операции загрузки данных и одну операцию записи данных за такт.

Выделенное функциональное устройство генерации адреса для записи данных немного проще в исполнении в сравнении с функциональными устройствами генерации адреса общего назначения (для записи и загрузки данных). Дело в том, что микрооперация записи данных просто записывает адрес (и, в конечном счете, сами данные) в буфер записи (store buffer). А микрооперация загрузки данных должна записывать в буфер чтения и также отслеживать содержимое буфера записи, для того чтобы исключить возможные конфликты.

Как только сгенерирован нужный виртуальный адрес, начинается просмотр кэша L1 DTLB на предмет соответствия этого виртуального адреса физическому. Сам кэш данных L1 DTLB в микроархитектуре Haswell не претерпел изменений.

При промахе в кэше L1 DTLB начинается просмотр соответствующих записей в унифицированном кэше L2 TLB, который имеет ряд улучшений в микроархитектуре Haswell.

Сам кэш данных L1 остался размером 32 Кбайт и 8-канальным (как и в микроархитектуре Sandy Bridge).

Однако в микроархитектуре Haswell кэш данных L1 имеет более высокую пропускную способность. Он поддерживает одновременно две 256-битных операций чтения и одну 256-битную операцию записи, что в совокупности дает агрегированную полосу пропускания в 96 байт за такт. В микроархитектуре Sandy Bridge кэш данных L1 поддерживает одновременно две 128-битных операций чтения и одну 128-битную операцию записи, то есть имеет теоретическую полосу пропускания в два раза ниже. При этом реальная полоса пропускания кэша данных L1 в микроархитектуре Sandy Bridge более чем вдвое ниже полосы пропускания в микроархитектуре Haswell по причине того, что в Sandy Bridge только два функциональных блока AGU.

Кроме того, в микроархитектуре Haswell увеличена и пропускная способность между кэшами L1 и L2. Так, если в Sandy Bridge пропускная способность между кэшем L2 и L1 составляла 32 байта за цикл, то в Haswell она повышена до 64 байтов за цикл. И при этом кэш L2 в Haswell имеет ту же латентность, что и в Sandy Bridge. В заключение отметим, что, как и в микроархитектуре Sandy Bridge, в Haswell кэш L2 не эксклюзивен и не инклюзивен по отношению к кэшу L1.

Графическое ядро в микроархитектуре Haswell

Одно из основных нововведений в микроархитектуре Haswell — это новое графическое ядро с поддержкой DirectX 11.1, OpenCL 1.2 и OpenGL 4.0 (рис. 3 17).

Но самое главное, что графическое ядро в микроархитектуре Haswell масштабируемое. Существуют варианты графического ядра с кодовыми названиями GT3, GT2 и GT1.

Ядро GT1 имеет минимальную производительность, а GT3 — максимальную.

В графическом ядре GT3 появился второй вычислительный блок, за счет чего удвоилось количество блоков растеризации, пиксельных конвейеров, вычислительных ядер и сэмплеров. Ожидается, что GT3 будет вдвое производительнее GT2.

Ядро GT3 содержит 40 исполнительных блоков, 160 вычислительных ядер и четыре текстурных блока. Для сравнения напомним, что в графическом ядре Intel HD Graphics 4000 процессоров Ivy Bridge содержится 16 исполнительных устройств, 64 вычислительных ядра и два текстурных блока. Поэтому, несмотря на приблизительно одинаковые тактовые частоты их работы, графическое ядро Intel GT3 превосходит своего предшественника по уровню производительности. Кроме того, ядро GT3 имеет более высокую производительность благодаря интеграции памяти EDRAM (в ядре GT3e) в упаковку процессора.

Ядро GT2 содержит 20 исполнительных блоков, 80 вычислительных ядер и два текстурных модуля, а ядро GT1 — только 10 исполнительных блоков, 40 вычислительных ядер и один текстурный модуль.

Сами исполнительные блоки имеют по четыре вычислительных ядра.

Еще одно нововведение заключается в том, что при работе с памятью применятся технология Instant Access, которая позволяет вычислительным ядрам процессора и графическому ядру напрямую обращаться к оперативной памяти. В предыдущих версиях графического ядра вычислительные ядра процессора и графическое ядро тоже работали с общей оперативной памятью, но при этом память делилась на две области с динамически изменяемыми размерами. Одна из них отводилась для графического ядра, а другая — для вычислительных ядер процессора.

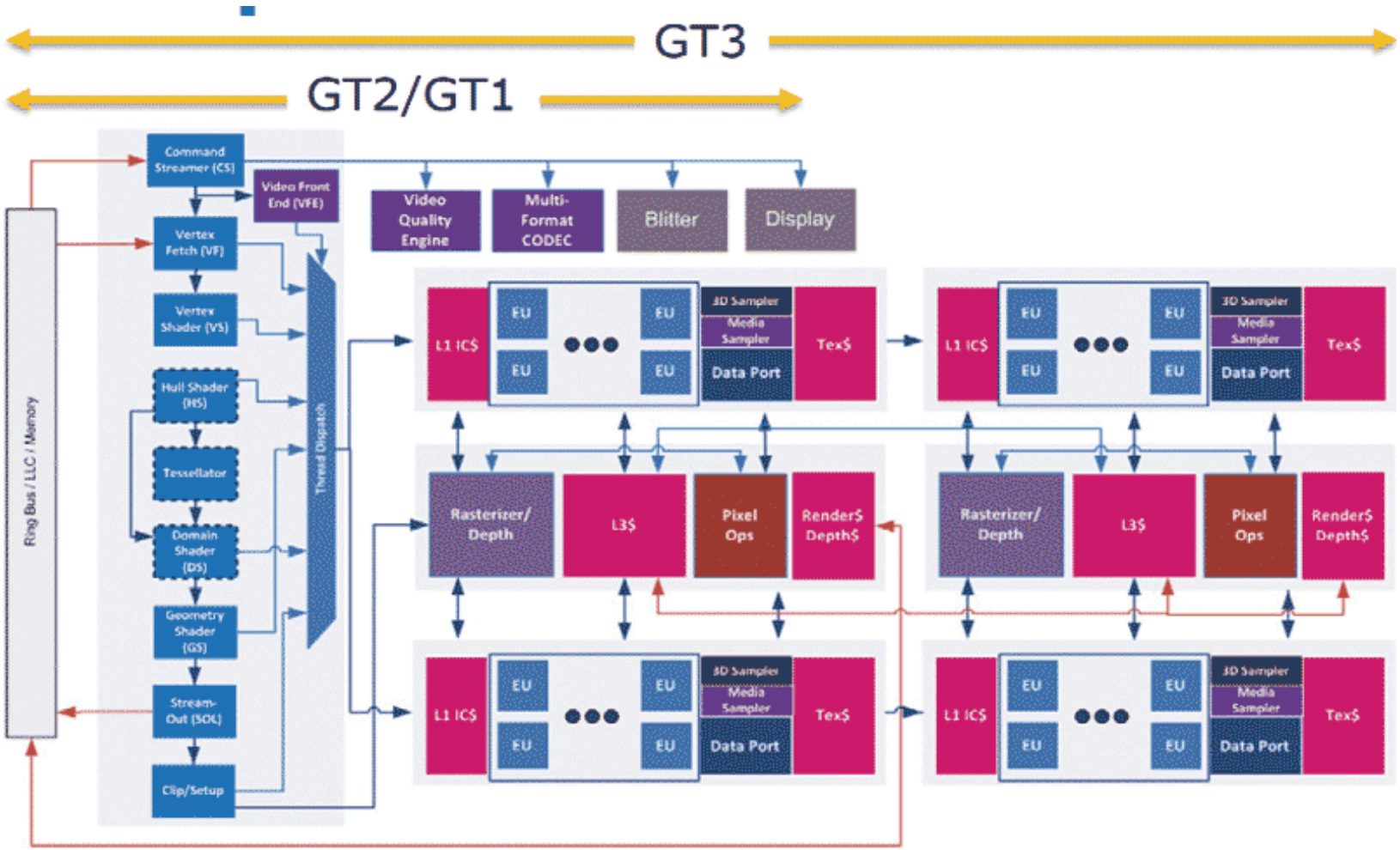


Рис. 3.17. Структура графического ядра

Технология InstantAccess позволяет драйверу графического ядра ставить указатель на положение определенного участка в области памяти графического ядра, к которой вычислительному ядру процессора необходимо напрямую получить доступ. При этом вычислительное ядро процессора будет работать с этой областью памяти напрямую, без создания копии, а после выполнения необходимых действий область памяти будет возвращена в распоряжение графического ядра.

Семейство новых графических ядер GT1, GT2 и GT3 обладает улучшенными возможностями по кодированию-декодированию видеоданных. Поддерживается аппаратное декодирование форматов H.264/MPEG-4 AVC, VC-1, MPEG-2, MPEG-2 HD, Motion JPEG, DivX с разрешением вплоть до 4096×2304 пикселов. Заявляется, что графическое ядро способно одновременно декодировать несколько видеопотоков 1080p и воспроизводить видео 2160p без подтормаживания и пропуска кадров.

Появился и специальный блок улучшения качества видео, который называется Video Quality Engine и отвечает за шумоподавление, цветокоррекцию, деинтерлейсинг, адаптивное изменение контраста и т.д. Также новые графические ядра будут поддерживать функции стабилизации изображения, преобразования частоты кадров и расширенной гаммы.

Кроме того, графическое ядро в процессоре Haswell обеспечивает подключение до трех мониторов одновременно.

3.4.7. Микроархитектура Skylake

Шестое поколение многоядерных процессоров Intel Core с рабочим названиемSkylake с полным на то правом можно назвать одним из наиболее масштабируемых и революционных за всю историю архитектуры Core. В этом заявлении нет ни малейшего преувеличения. Так, масштабируемость подтверждает ассортимент из почти 50 наименований Xeon, Core i3/5/7, Core M3/5/7, Pentium и Celeron с впечатляющим разбросом характеристик: от крохотных (20 x 16,5 мм) чипов в компактной корпусировке BGA1515 с TDP 4,5 Вт до мощных разблокированных десктопных LGA1151 процессоров вроде Core i7-6700K с габаритами 37,5 x 37,5 мм и TDP порядка 91 Вт. То есть, 20-кратная масштабируемость по энергопотреблению и 4-кратная по размерам чипа.

Кроме того, процессоры с архитектурой Skylake, выпускаемые с соблюдением норм 14-нм техпроцесса Intel, появятся быстро, и сразу практически для всех сегментов вычислительной техники – от мобиль-

ных устройств до серверов. Это гораздо энергичнее, нежели 22-нм чипы Haswell двухлетней давности, и гораздо масштабнее чем предыдущее поколение с рабочим названием Broadwell, когда, почти "в обход" десктопных платформ, основной упор был сделан на чипы для ноутбуков и планшетов.

Что касается тезиса о революционности, он подтверждается действительно существенным изменением схемотехники и производительности большинства ключевых элементов архитектуры, таких как DDR4/DDR3L, eDRAM, графика HD Intel Graphics 5xx, Iris/Iris Pro и многое другое, чему, соответственно, и посвящён этот материал.

Архитектура процессоров Skylake получила сотни структурных изменений и улучшений, позволивших повысить производительность при снижении потребления энергии.

Вычислительные процессорные ядра при этом, хоть и изменились, но остались примерно сравнимыми по структурному строению с предыдущими поколениями Broadwell и Haswell. Чего нельзя сказать о структурном уровне чипа в целом (рис. 3.18), где появился ряд совершенно новых модулей и блоков. Благодаря этому новые чипы Intel Core шестого поколения окончательно превратились из классических процессоров в так называемые "системы на чипе" (SoC, System on Chip).

Стандартный набор компонентов процессора Skylake состоит из двух или четырёх вычислительных ядер (CPU), графической подсистемы, общей кольцевой коммуникационной шины, блока платформенных контроллеров PCH (Platform Controller Hub, порой до сих пор называемого "южным мостом") на многоканальной шине DMI/OPI, интегрированного "расширителя кэша" eDRAM (бывший Crystalwell, опциональный у Haswell), шины PCI Express x16, а также встроенного модуля системных блоков System Agent. В свою очередь, в состав System Agent входят как уже привычные (но значительно переработанные), так и совершенно новые блоки, включая доработанный управляющий блок PCU (Package Control Unit) с блоком контроля температуры (PECI, Platform Environment Control Interface) и напряжения (SVID, Serial Voltage Identification), контроллер памяти DDR3L/DDR4, блоки мультимедийной обработки и вывода видео, плюс совершенно новый процессор обработки изображений ISP (Image Signal Processing).

Основным усовершенствованием процессорных x86-ядер Skylake, позволяющим говорить о повышении качества предсказания ветвлений, загрузки исполнительного конвейера и, как следствие, более частого одновременного декодирования CISC-инструкций и исполнения до шести микроинструкций за каждый такт, стоит назвать значительно рас-

ширенные по сравнению с предыдущими поколениями внутренние буферы.

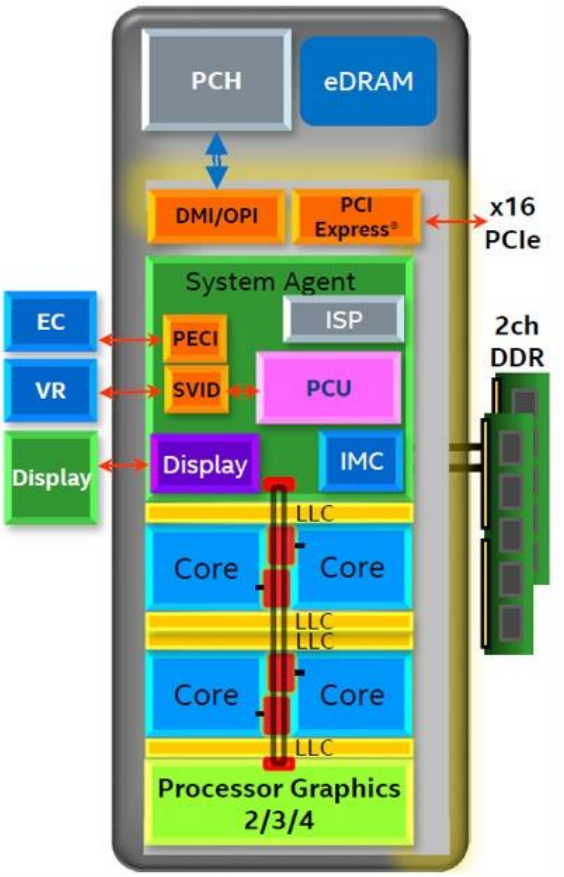


Рис. 3.18. Стандартный набор компонентов процессора

На представленном ниже скриншоте из презентации Intel наглядно виден последовательный рост емкости буферов на примере трёх последних поколений архитектуры Core (рис. 3.19).

	Sandy Bridge	Haswell	SkyLake
Out-of-order Window	168	192	224 ↑
In-flight Loads	64	72	72
In-flight Stores	36	42	56 ↑
Scheduler Entries	54	60	97 ↑
Integer Register File	160	168	180 ↑
FP Register File	144	168	168
Allocation Queue	28/thread	56	64/thread ↑

Рис. 3.19. Последовательный рост ёмкости внутренних буферов процессора

Обработка входящих команд улучшена благодаря ускоренной работе более ёмкого блока предсказания ветвлений, более глубокая буферизация внеочередного исполнения инструкций позволила говорить о более качественном распараллеливании обрабатываемого кода, уменьшении латентности и снижении энергопотребления в моменты простоя (рис. 3.20).

В целом архитектура Skylake обладает более глубокой буферизацией данных, чтения/записи, отложенной (write-back) записи, ускоренной производительностью обработки промахов страниц и кэш-памяти L2. Со времён Sandy Bridge почти в полтора раза (до 224) увеличилось окно внеочередного исполнения инструкций, улучшена работа Hyper-Threading за счёт почти удвоенного до 64 на поток окна очереди распределения и сниженного простоя конвейера за счёт оптимизированных алгоритмов улучшенного блока предсказания ветвления. Также появились новые инструкции для лучшего управления загрузкой кэш-памяти, а скорость работы протокола AES криптографического шифрования в GCM- и CBC-режимах выросла на 17% и 33%, соответственно.

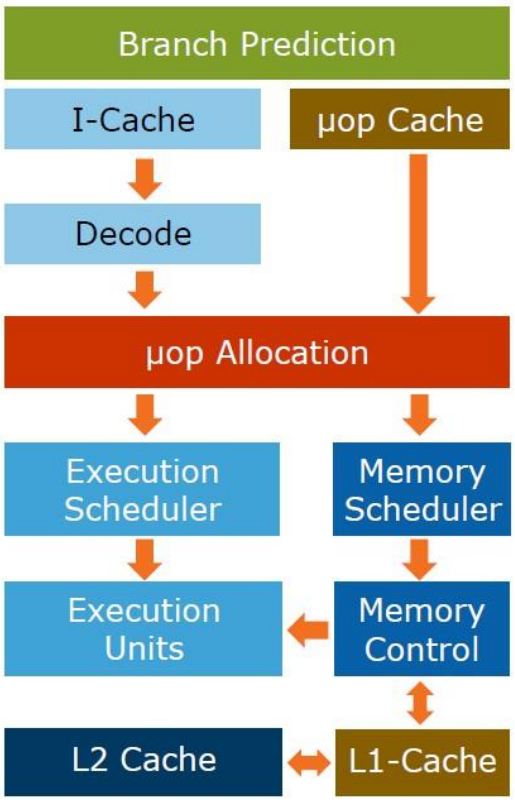


Рис. 3.20. Структура ядра процессора

В архитектуре Skylake появилась поддержка наборов новых инструкций MIC и SGX.

Серьёзным изменениям в архитектуре Skylake подверглась кольцевая шина, кэш-память и структура цепей работы с памятью. Согласно данным Intel, пропускная способность кольцевой шины, обеспечивающей обмен между вычислительными ядрами, графической подсистемой, системным агентом, контроллером памяти и кэш-памятью L3 была удвоена по сравнению с поколением Haswell, при этом число используемых для этого транзисторов увеличилось лишь на 50%, а уровень энергопотребления для многих режимов остался на прежнем уровне.

Удвоенная производительность кольцевой шины позволила удвоить скорость работы кэш-памяти L3 при обработке промахов, что вместе с появлением поддержки DDR4 и особенностями работы eDRAM даёт надежду на значительный прирост производительности в некоторых приложениях. Помимо этого улучшенная работа с памятью также сказалась на обеспечении стабильной работы процессора обработки изображений (ISP) и поддержке видеовыхода на три дисплея с разрешением до 4K.

В архитектуре Skylake реализована новая, полностью когерентная структура встроенной DRAM (eDRAM), или Memory Side Cache, способная кэшировать любые данные, включая варианты "некэшируемой памяти", без необходимости очистки для поддержания когерентности, и доступной для использования устройствами ввода-вывода и формирования выходного видеосигнала. Помимо этого графическая подсистема для достижения оптимальной производительности может выбрать режим кэширования определённых данных только в eDRAM без использования кэш-памяти L3.

В отличие от предыдущей архитектуры, где примерно четверть кэш-памяти L3 использовалась для доступа к eDRAM, и при этом eDRAM не имела возможности прямого взаимодействия с остальной системой, в архитектуре Skylake контроллер eDRAM переместился в модуль системного агента, освободив таким образом порядка 512 Кбайт ёмкости кэша L3 и одновременно с этим облегчив доступ другим компонентам ядра к данным в eDRAM. Отныне Memory Side Cache может взаимодействовать с основной системной памятью напрямую, обеспечивая таким образом обновление экрана без необходимости вывода остальных компонентов процессора из ждущего режима.

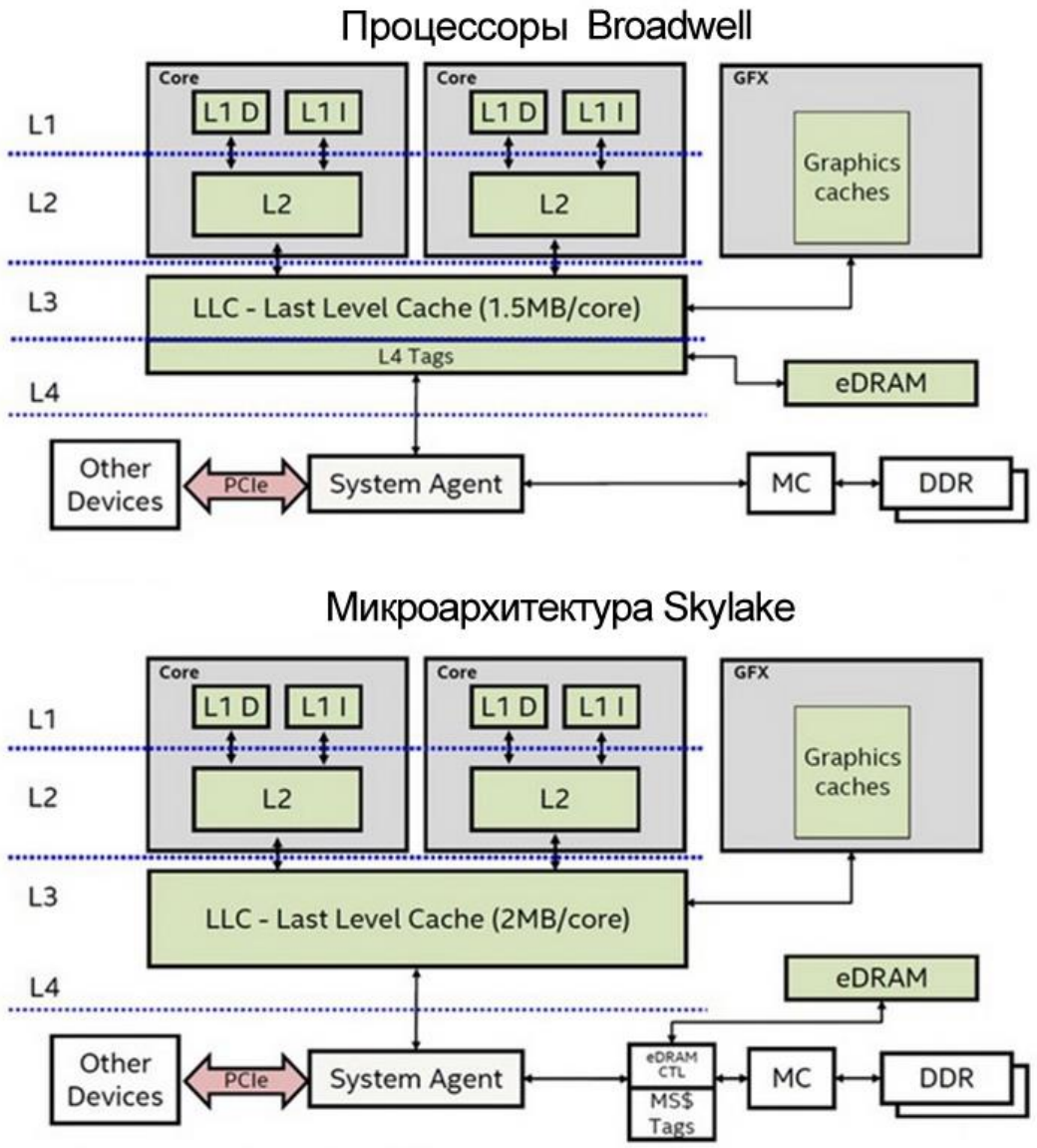


Рис. 3.21. Конфигурация процессора с eDRAM

К сожалению, как и в случае с архитектурой Haswell, в Intel так и не озвучили планов использования eDRAM в составе обычных LGA-процессоров на базе Skylake для настольных систем. Не исключено, что на этой стадии развития процессорной микроархитектуры Intel плюсы Memory Side Cache будут реализованы только в чипах для мобильных, встраиваемых и компактных систем.

Полупроводниковая логика и программная прошивка управляющего блока PCU в архитектуре Skylake подверглась значительным доработкам для достижения более динамичных режимов с высокими уровнями производительности и агрессивными алгоритмами энергосбере-

жения, в том числе, в плане сбора внутренней статистики, внешней и внутренней телеметрии (iMon, Psys) и более тесного взаимодействия со старшими в иерархии системами управления (OS, BIOS).

Самое заметное и, пожалуй, существенное изменение в архитектуре Skylake связано с новым интегрированным контроллером оперативной памяти: он по-прежнему 2-канальный, однако теперь поддерживает динамическую память как предыдущего стандарта DDR3L, так и нового поколения DDR4. Прежде полноценный контроллер DDR4 был на вооружении только серверных чипов Xeon и выполненных на основе их дизайна топовых геймерских LGA 2011 чипах Intel Core i7, но фактически именно начиная со Skylake память DDR4 начинает своё широкомасштабное наступление на рынок настольных и мобильных систем.

Модуль аппаратной обработки мультимедийного контента и финального рендеринга видеопотока в архитектуре Skylake также значительно доработан и улучшен. В частности, аппаратный кодек HEVC/H265 нового поколения обладает повышенной производительностью с одновременным уменьшением энергопотребления; есть отдельный аппаратный кодек AVC с очень низким энергопотреблением; обеспечивается вывод изображения на три независимых 4K-дисплея со сниженным на 40%-60% потреблением энергии даже в самом тяжёлом режиме воспроизведения 4K-видео.

В дополнение, архитектура Skylake также поддерживает инициативу Intel по отказу от проводных подключений для беспроводной передачи мультимедийного контента с помощью технологий Intel WiDi или Pro WiDi с компьютеров на телевизоры, мониторы или проекторы.

Впервые в составе архитектуры для массовых процессоров Skylake (а не специализированных SoC) появился так называемый встроенный процессор обработки изображений – ISP (Image Signal Processing), что особенно актуально для смартфонов, планшетов и ноутбуков. В частности, ISP обладает встроенным интерфейсом CSI (Camera Sensor Interface) с поддержкой до четырех внешних цифровых камер/сенсоров с разрешением до 13 Мп, правда, с одновременным обслуживанием только двух из них. Аппаратная обвязка CSI поддерживает расширенный список технологий для полноценной обработки фото и видео, включая распознавание и запоминание лиц, групповые снимки, многопоточный захват, съёмку с расширенным динамическим диапазоном (HDR), съёмку при слабом освещении, серийную съёмку и многое другое.

Роль графических ядер, встроенных в процессоры, с каждым годом увеличивается. И это связано не столько с ростом их 3D-

производительности, столько с тем, что встроенные GPU берут на себя всё новые функции, такие как параллельные вычисления или кодирование и декодирование мультимедийного контента. Исключением не стало и графическое ядро Skylake. Intel относит его к следующему, девятому поколению, и это значит, что в нём таится немало сюрпризов. Однако начать стоит с того, что GPU, реализованный в Skylake, как и его предшественники, сохранил традиционный модульный дизайн. Таким образом, мы вновь имеем дело с целым семейством решений разного класса: на базе имеющихся строительных блоков нового поколения Intel может собирать кардинально различающиеся по уровню производительности GPU. Подобная масштабируемость сама по себе новинкой не является, но в Skylake возросла не только максимальная производительность, но и число доступных вариантов графического ядра.

Итак, графическое ядро Skylake может быть построено на базе одного или нескольких модулей, каждый из которых обычно включает в себя по три секции. Секции объединяют по восемь исполнительных устройств, на которые ложится основная часть обработки графических данных, а также содержат базовые блоки для работы с памятью и текстурные семплеры. Помимо исполнительных устройств, сгруппированных в модули, графическое ядро содержит и внемодульную часть, отвечающую за фиксированные геометрические преобразования и отдельные мультимедийные функции.

На самом верхнем уровне иерархии графическое ядро Skylake очень похоже на ядро, реализованное в Broadwell. Однако если углубиться в подробности, то нетрудно найти и заметные изменения.

Во-первых, внемодульная часть вынесена теперь в отдельный энергетический домен, что позволяет задавать ей частоту и отправлять её в сон отдельно от исполнительных устройств. Это значит, что, например, при работе с технологией Quick Sync, которая реализуется как раз силами внемодульных блоков, основная часть GPU может быть отключена от линий питания в целях снижения энергопотребления. Кроме того, независимое управление частотой внемодульной части позволяет лучше подстраивать её производительность под конкретные нужды модулей графического ядра.

Во-вторых, в то время как графическое ядро Broadwell могло основываться лишь на одном или двух модулях, получая в своё распоряжение 24 или 48 исполнительных устройств (для энергоэффективных и бюджетных процессоров мог использоваться один модуль с отключенными секциями, что давало меньшее, чем 24, число исполнительных устройств), в Skylake может применяться от одного до трёх модулей.

Благодаря этому в дополнение к привычным конфигурациям GT1/GT2/GT3 в семействе процессоров Skylake будет доступно ещё более мощное ядро GT4 (рис. 3.22), которое получит 72 исполнительных устройства.

Однако пиковая производительность самих исполнительных устройств в Skylake не изменилась – каждое такое устройство может выполнять до 16 32-битных операций за такт. При этом оно способно исполнять 7 вычислительных потоков одновременно и имеет 128 32-байтовых регистров общего назначения.

В-третьих, варианты ядра GT3 и GT4 могут быть дополнительно усилены eDRAM-буфером объёмом 64 или 128 Мбайт соответственно, что даёт модификации GT3e и GT4e. Процессоры Broadwell комплектовались лишь одним вариантом eDRAM – объёмом 128 Мбайт. В Skylake же этот дополнительный буфер не только изменил алгоритм работы, став «кешем на стороне памяти», но и приобрёл некоторую гибкость конфигурации. Однако его исполнение останется старым – он будет представлен отдельным 22-нм кристаллом, монтируемым на процессорную плату по соседству с основным чипом.

3.4.8. Микропроцессоры семейства «Эльбрус»

В 1969 году в связи с необходимостью в высокопроизводительной вычислительной технике для оснащения в нашей стране стратегических систем специального назначения возникла идея разработки многопроцессорных вычислительных комплексов (МВК) семейства «Эльбрус». Основоположником идеи был Лебедев С.А., а главным конструктором прокта в Институте точной механики и вычислительной техники (ИТМ и ВТ) стал Бурцев В.С.

Разработка МВК «Эльбрус-1» началась в 1973 г., закончена в 1979 г. и сдана государственной комисси в 1980 г. Данный вычислительный комплекс был построен на базе TTL-микросхем.

Затем был построен МВК «Эльбрус-2». Годы разработки 1977-1984, сдан в 1985 г. Данный комплекс в течении нескольких лет использовался в центральных объектах стратегических систем страны.

Следующим проектом стал МВК «Эльбрус-3». Руководителем проекта был Бабаян Б.А. Опытный образец данного комплекса был изготовлен в 1990 году, но в связи с прекращением финансирования отладка образца не была завершена.

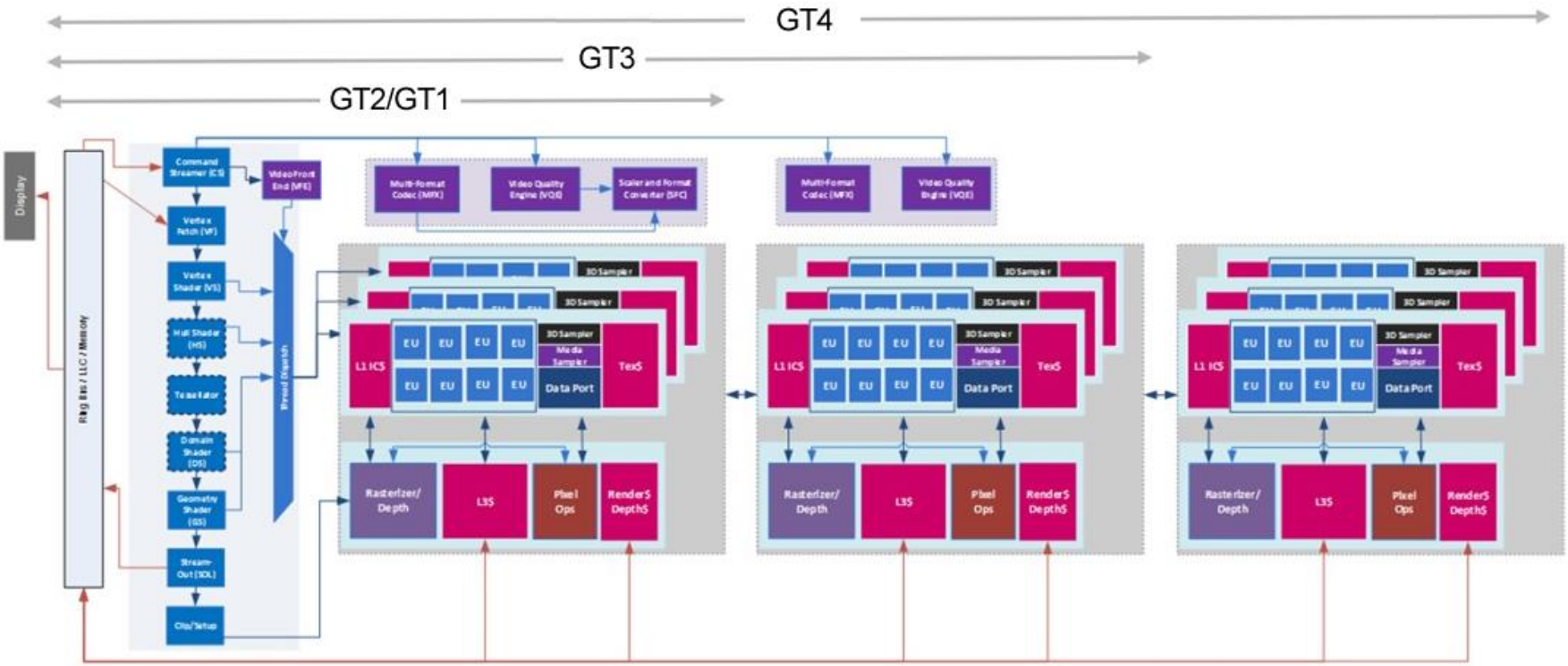


Рис. 3.22. Структура графического ядра Skylake

Стало ясно, что будущие поколения вычислительной техники должны базироваться на микропроцессорах. В 1992 г. коллектив разработчиков машин семейства «Эльбрус» выделился в компанию ЗАО «МЦСТ – Московский центр SPARC-технологий» и совместно с ОАО «ИНЭУМ имени И.С. Брука» начал вести работы над микропроцессорной реализацией разработанной архитектуры.

Основой серии осталась оригинальная архитектура «Эльбрус», а также добавилась открытая архитектура SPARC «Scalable Processor Architecture» (табл. 3.2).

Таблица 3.2

Разработка линии микропроцессоров «Эльбрус»

Микропроцессор	«Эльбрус»	«Эльбрус-2С»	«Эльбрус-4С»	«Эльбрус-8С»
Технологическое разрешение, нм	130	90	65	28
Год выпуска	2007	2010	2012	2014
Чисто процессорных ядер	1	2	4	8
Частота, МГц	300	600	800	1300
Производительность, GFlops	4,8	19,2	50	250
Мощность, Вт	6	16	25	60

Кроме того, появилось семейство микропроцессоров МЦСТ-R. Это семейство всключает в себя микропроцессоры: МЦСТ-R150, МЦСТ-R500, МЦСТ-R500S, МЦСТ-R1000 и МЦСТ-R1000M. Все микросхемы реализуют архитектуру SPARC.

Среди особенностей архитектуры «Эльбрус» инженеры МЦСТ выделяют следующие:

- 6 каналов арифметико-логических устройств, работающих параллельно;
- регистровый файл из 256 84-разрядных регистров;
- аппаратная поддержка циклов, в том числе с конвейеризацией;
- программируемое асинхронное устройство предварительной подкачки данных с отдельными каналами считывания;
- поддержка спекулятивных вычислений и однобитовых предикатов;
- широкая команда (VLIW-архитектура), способная при максимальном заполнении задать в одном такте до 23 операций (более 33 операций при упаковке операндов в векторные команды).

Кроме того, в архитектуре присутствует режим x86-совместимости. Для этого реализована система двоичной трансляции двоичных кодов x86 в коды процессра «Эльбрус».

Микропроцессор «Эльбрус-8С»

В июне 2014 г. была запущена в производство опытная партия универсальных микропроцессоров «Эльбрус-8С». Данный микропроцессор – это полностью российская разработка. В таблице 3.3 приведены основные характеристики микропроцессора «Эльбрус-8С».

Таблица 3.3

Технические характеристики

Параметр	Значение	Примечание
Архитектура процессора	«Эльбрус»	Количество вычислительных устройств с плавающей запятой увеличено с 4 до 6
Технологический процесс	28 нм	
Рабочая тактовая частота	1300 МГц	Расчетное значение
Количество ядер	8	
Производительность	250 GFlops	На операциях с одинароной точностью (FP32)
Кэш-память 2-го уровня	8 x 512 Кбайт	Отдельная кэш-память для каждого ядра
Кэш-память 3-го уровня	16 Мбайт	Разделяемая между всеми ядрами
Тип контроллеров памяти	DDR3-1600	С поддержкой ECC
Количество контроллеров памяти	4	
Поддержка многопроцессорных систем	До 4-х процессоров	
Каналы межпроцессорного обмена (пропускная способность)	3 (16 Гбайт/с)	Каналы дуплексные (пропускная способность в каждую сторону – 8 Гбайт/с)

Базовой операционной системой для платформы «Эльбрус» является ОС «Эльбрус», построенная на базе ядра Linux. Система программирования платформы поддерживает языки C, C++, Java, Фортран-77, Фортран-90.

На базе микропроцессора «Эльбрус-8С» планируется организовать массовое производство серверов, рабочих станций и других средств вычислительной техники, предназначенных для применения в государсвунных учреждениях и бизнес-структурах, предъявляющих повышенные требования к информационной безопасности, а также для применения в области высокопроизводительных вычислений, обработки сигналов, телекоммуникации.

3.4.9. Микропроцессоры IBM POWER8

В агусте 2013 г. IBM представила восьмое поколение своих RISC-процессоров POWER, которые используются в её Unix-серверах. Если предыдущее поколение POWER7 было восьмиядерном, то новый POWER8 может содержать до двенадцати процессорных ядер. Максимальная тактовая частота – до 5 ГГц.

POWER8 разработан как микропроцессор с расширенной поддержкой многопоточности (рис. 3.23). Так, каждое ядро POWER8 имеет аппаратную поддержку одновременного исполнения до 8-ми потоков, следовательно, 12-ядерный микропроцессор поддерживает до 96 потоков. В процессоре используется значительное количество eDRAM памяти в качестве кэшей (как на пластине процессора, так и вне её). На каждое ядро выделены кэши L1 размером 64 и 32 Кб (данные и инструкции), кэши L2 размером 512 Кб; процессор также имеет общий кэш L3 размером 48 (6-ядерные модели) или 96 Мб (12-ядерные модели). В процессор встроены высокопроизводительные контроллеры памяти (DDR3/DDR4) и системных каналов ввода-вывода (CAPI port на основе PCI Express 3.0 в том числе, для подключения ASIC, FPGA, GPU). Питанием процессора управляет встроенный микроконтроллер на базе PowerPC 405 с 512 килобайтами SRAM памяти, настраивая 1764 встроенных регуляторов напряжения.

Для многих видов нагрузок процессор POWER8 показывает прирост производительности в 2-3 раза по сравнению с предыдущим процессором POWER7.

Производится по техпроцессу 22 нм по технологии кремний на изоляторе с 15 слоями металлизации. Двенадцатиядерный вариант содержит 4,2 миллиарда транзисторов и имеет площадь кристалла в 650 мм^2. Шестиядерный вариант меньше — всего 362 мм^2.

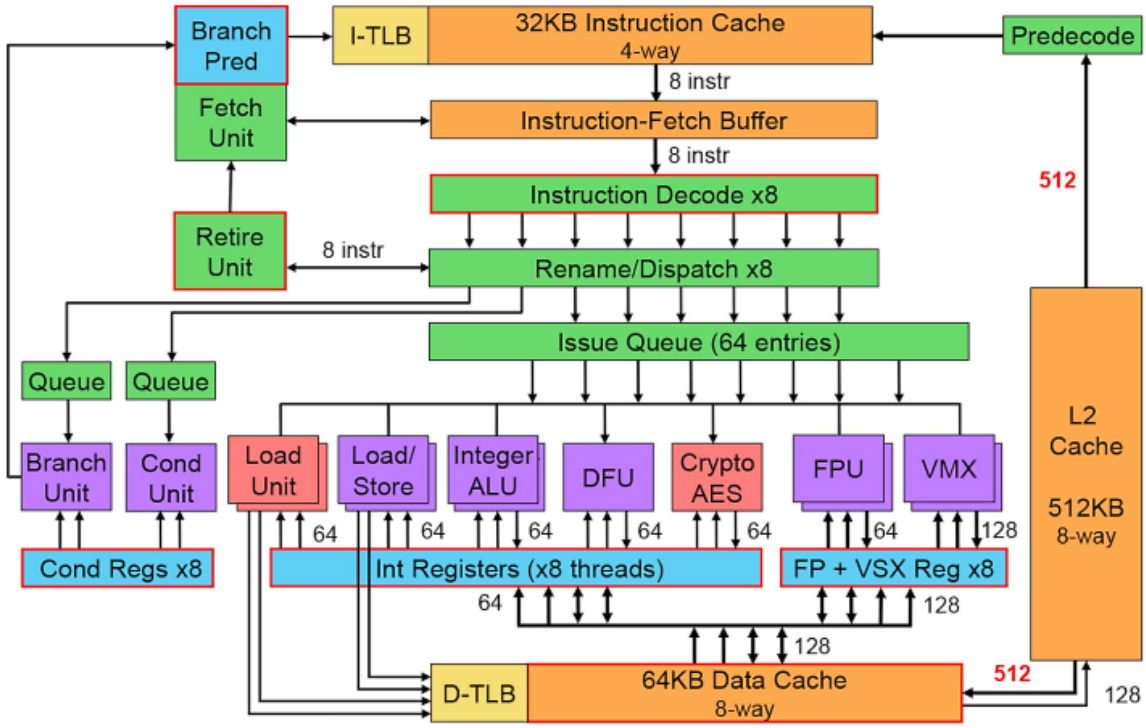


Рис. 3.23. Структура ядра микропроцессора POWER8

Серверы на базе POWER8 начали выпускаться с июня 2014 г. при поддержке консорциума OpenPower. В консорциум входят такие корпорации, как IBM, Google, Nvidia и др. (всего 25 компаний). Отметим, что будучи членом OpenPower Foundation, IBM в этот раз предоставила членам консорциума детальную информацию и спецификации своих новых процессоров. Впервые IBM предложила лицензирование высокопроизводительного ядра архитектуры POWER. Устройство микропроцессоров будет доступно для покупки другими компаниями, входящими в OpenPower.

4. ПРИНЦИПЫ ОРГАНИЗАЦИИ ПОДСИСТЕМЫ ПАМЯТИ ЭВМ И ВС

4.1. Иерархическая структура памяти ЭВМ

Памятью ЭВМ называется совокупность устройств, предназначенных для запоминания, хранения и выдачи информации.

Основными характеристиками отдельных устройств памяти (запоминающих устройств) являются емкость памяти, быстродействие и стоимость хранения единицы информации (бита).

Емкость памяти определяется максимальным количеством данных, которые могут в ней храниться. Ёмкость измеряется в двоичных единицах (битах), машинных словах, но большей частью в байтах (1 байт = 8 бит). Часто емкость памяти выражают через число К = 2^{10} = 1024, например, 1024 бит = Кбит (килобит), 1024 байт = Кбайт (килобайт), 1024 Кбайт = 1 Мбайт (мегабайт), 1024 Мбайт = 1 Гбайт (гигабайт), 1024 Гбайт = 1 Тбайт (терабайт).

Быстродействие (задержка) памяти определяется **временем доступа** и **длительностью цикла памяти**. Время доступа представляет собой промежуток времени между выдачей запроса на чтение и моментом поступления запрошенного слова из памяти. Длительность цикла памяти определяется минимальным временем между двумя последовательными обращениями к памяти.

Требования к увеличению емкости и быстродействия памяти, а также к снижению ее стоимости являются **противоречивыми.** Чем больше быстродействие, тем технически труднее достигается и дороже обходится увеличение емкости памяти. Исходя из этого, память ЭВМ организуется в виде **иерархической структуры** запоминающих устройств, обладающих различным быстродействием, емкостью и стоимостью. Причем более высокий уровень меньше по объему, быстрее и имеет большую стоимость в пересчёте на байт, чем более низкий уровень. Уровни иерархии взаимосвязаны: все данные на одном уровне могут быть также найдены на низком уровне, и все данные на этом (более низком) уровне могут быть найдены на следующем, нижележащем уровне, и так далее, пока мы не достигнем основания иерархии. В структуре памяти, представленной на рис. 4.1, к верхнему (сверхоперативному) уровню относятся: управляющая память, регистры различ-

ного назначения, стек регистров, буферная память. На втором уровне находится основная (или оперативная) память. На последующих уровнях размещается внешняя и архивная память. Система управления памятью обеспечивает обмен информационными блоками между уровнями, причем обычно первое обращение к блоку информации вызывает его перемещение с низкого (медленного) уровня на более высокий. Это позволяет при последующих обращениях к данному блоку осуществлять его выборку с более быстродействующего уровня памяти.

Успешное или неуспешное обращение к более высокому уровню называется соответственно «попаданием» (hit) или «промахом» (miss). Попадание есть обращение к объекту в памяти, который найден на более высоком уровне, в то время как промах означает, что он не найден на этом уровне. Доля попаданий, или коэффициент попаданий, есть доля обращений, найденных на более высоком уровне. Иногда она представляется в процентах. Аналогично для промахов.

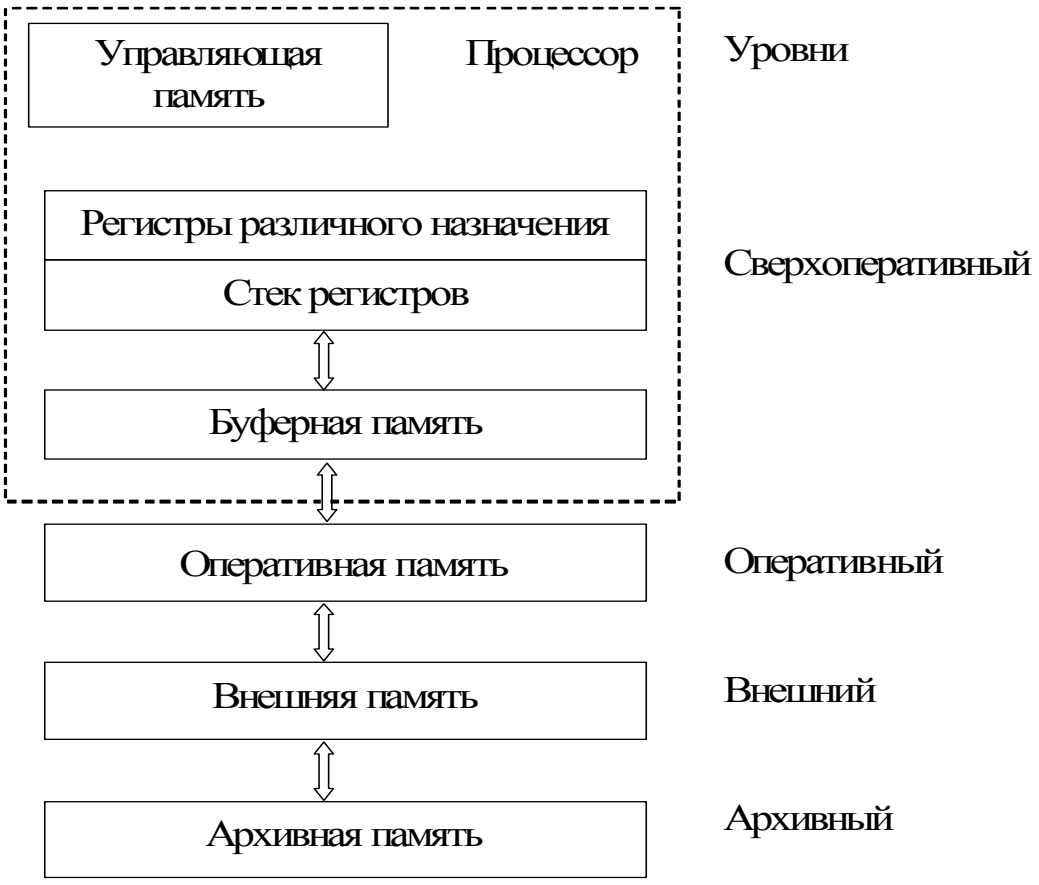


Рис. 4.1. Иерархическая структура памяти

Сравнительно небольшая емкость оперативной памяти компенсируется практически неограниченной емкостью внешних запоминающих устройств. Однако эти устройства работают намного медленнее, чем оперативная память. Время обращения за данными для магнитных дисков составляет десятки микросекунд. Для сравнения: цикл обращения к оперативной памяти (ОП) составляет несколько десятков наносекунд. Исходя из этого, вычислительный процесс должен протекать с возможно меньшим числом обращений к внешней памяти. Память современных компьютеров реализуется на микросхемах статических и динамических запоминающих устройств с произвольной выборкой. Микросхемы статических ЗУ (SRAM) имеют меньшее время доступа и не требуют циклов регенерации (восстановления) информации. Микросхемы динамических ЗУ (DRAM) характеризуются большей емкостью и меньшей стоимостью, но требуют схем регенерации и имеют значительно большее время доступа. У статических ЗУ время доступа совпадает с длительностью цикла.

По этим причинам в основной памяти практически любого компьютера, проданного после 1975 г., использовались полупроводниковые микросхемы DRAM (SDRAM, DDR SDRAM, RDRAM). Для построения кэш-памяти применяются SRAM.

Непрерывный рост производительности ЭВМ проявляется в первую очередь в повышении скорости работы процессора. Быстродействие ОП также растет, но все время отстает от быстродействия аппаратных средств процессора в значительной степени потому, что одновременно происходит опережающий рост её емкости, что делает более трудным уменьшение времени цикла работы памяти. Вследствие этого быстродействие ОП часто оказывается недостаточным для обеспечения требуемой производительности ЭВМ. Это проявляется в несоответствии пропускных способностей процессора и ОП. Возникающая проблема выравнивания их пропускных способностей решается путем использования сверхоперативной буферной памяти небольшой емкости и повышенного быстродействия, хранящей команды и данные, относящиеся к обрабатываемому участку программы.

При обращении к блоку данных, находящемуся на оперативном уровне, его копия пересылается в сверхоперативную буферную память (СБП). Последующие обращения производятся к копии блока данных, находящейся в СБП. Поскольку время выборки из сверхоперативной буферной памяти $t_{СБУ}$ (несколько наносекунд) много меньше времени выборки из оперативной памяти $t_{ОП}$, введение в структуру памяти СБП

приводит к уменьшению эквивалентного времени обращения $t_Э$ по сравнению с $t_{ОП}$:

$$t_Э = t_{СБП} + \alpha t_{ОП},$$

где $\alpha = (1 - q)$ и q – вероятность нахождения блока в СБП в момент обращения к нему, т.е. вероятность «попадания». Очевидно, что при высокой вероятности попадания эквивалентное время обращения приближается к времени обращения к СБП.

В основе такой организации взаимодействия ОП и СБП лежит принцип локальности обращений, согласно которому при выполнении какой-либо программы (практически для всех классов задач) большая часть обращений в пределах некоторого интервала времени приходится на ограниченную область адресного пространства ОП, причем обращения к командам и элементам данных этой области производятся многократно. Это позволяет копии наиболее часто используемых участков программ и некоторых данных загрузить в СБП и таким образом обеспечить высокую вероятность попадания q. Высокая эффективность применения СБП достигается при $q \geq 0{,}9$.

Буферная память не является программно доступной. Это значит, что она влияет только на производительность ЭВМ, но не должна оказывать влияния на программирование прикладных задач. Поэтому она получила название **кэш-памяти** (в переводе с английского – тайник). В современных компьютерах применяют многоуровневую кэш-память (до трех уровней), которая еще больше способствует производительности ЭВМ. Как правило, на первом уровне используются раздельные кэш-памяти для команд и данных, а на других уровнях данные и команды хранятся в одних и тех же кэш-памятях.

4.2. Организация стека регистров

Регистровая структура процессора была рассмотрена в разд. 3.

Стек регистров, реализующий безадресное задание операндов, является эффективным элементом архитектуры ЭВМ. Стек представляет собой группу последовательно пронумерованных регистров, снабжённых указателем стека, в котором автоматически при записи устанавливается номер первого свободного регистра стека (вершина стека). Существует два основных способа организации стека регистров:

LIFO (Last-in First-Out) – последний пришёл – первый ушёл;

FIFO (First-in First-Out) – первый пришёл – первый ушёл.

Механизм стековой адресации для первого способа поясняется на рис. 5.2. Для реализации адресации по способу LIFO используется счётчик адреса СЧА, который перед началом работы устанавливается в состояние ноль, и память (стек) считается пустой. Состояние СЧА определяет адрес первой свободной ячейки. Слово загружается в стек с входной шины *X* в момент поступления сигнала записи ЗП.

По сигналу ЗП слово *X* записывается в регистр Р[СЧА], номер которого определяется текущим состоянием счётчика адреса, после чего с задержкой *D*, достаточной для выполнения микрооперации записи Р[СЧА]:=*X*, состояние счетчика увеличивается на единицу. Таким образом, при последовательной загрузке слова *A*, *B* и *C* размещаются в регистрах с адресами Р[*S*], Р[*S* + 1] и Р[*S* + 2], где *S* – состояние счётчика на момент начала загрузки. Операция чтения слова из ЗУ инициируется сигналом ЧТ, при поступлении которого состояние счётчика уменьшается на единицу, после чего на выходную шину *Y* поступает слово, записанное в стек последним. Если слова загружались в стек в порядке *A*, *B*, *C*, то они могут быть прочитаны только в обратном порядке: *C*, *B*, *A*.

В современных архитектурах процессоров стек и стековая адресация широко используются при организации переходов к подпрограммам и возврата из них, а также в системах прерывания.

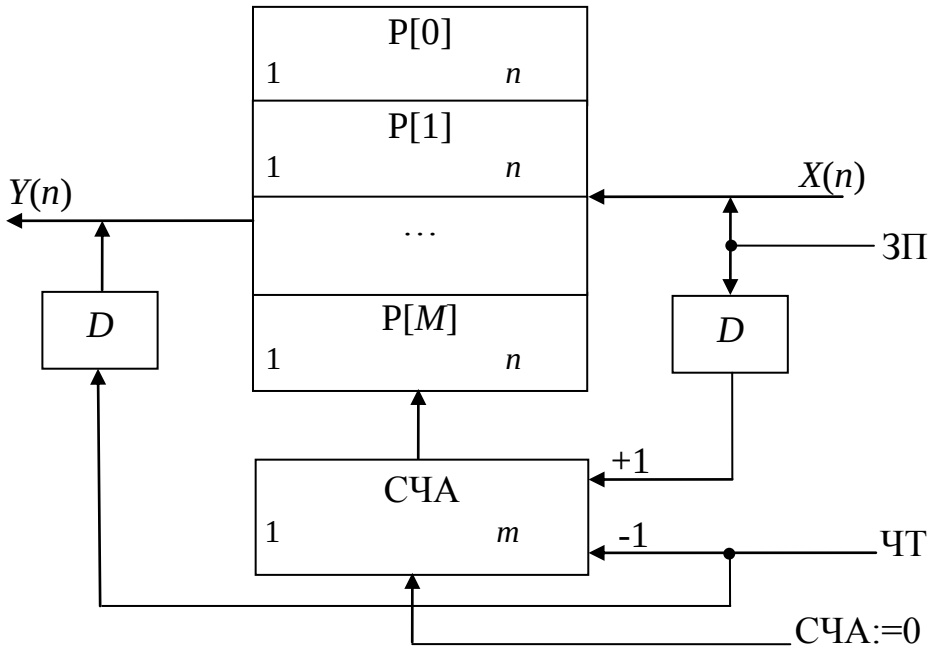


Рис. 4.2. Механизм стековой адресации по способу LIFO

4.3. Способы организации кэш-памяти

Общие сведения

В функциональном отношении кэш-память рассматривается как буферное ЗУ, размещённое между основной (оперативной) памятью и процессором. Основное назначение кэш-памяти – кратковременное хранение и выдача активной информации процессору, что сокращает число обращений к основной памяти, скорость работы которой меньше, чем кэш-памяти.

За единицу информации при обмене между основной памятью и кэш-памятью принята строка (линейка), причём под строкой понимается набор слов, выбираемый из оперативной памяти при одном к ней обращении. Хранимая в оперативной памяти информация представляется, таким образом, совокупностью строк с последовательными адресами. В любой момент времени строки в кэш-памяти представляют собой копии строк из некоторого их набора в ОП, однако расположены они необязательно в такой же последовательности, как в ОП.

Построение кэш-памяти может осуществляться по различным принципам, которые будут рассмотрены ниже. Но общим для всех способов построения кэш-памяти является использование так называемых адресных тегов. **Адресный тег** – это расширенный адрес, который объединяет адреса всех слов, принадлежащих данной строке. Он указывает, какую строку в ОП представляет данная строка в кэш-памяти.

4.3.1. Типовая структура кэш-памяти

Рассмотрим типовую структуру кэш-памяти (рис. 4.3), включающую основные блоки, которые обеспечивают её взаимодействие с ОП и центральным процессором.

Строки, составленные из информационных слов, и связанные с ними адресные теги хранятся в накопителе, который является основой кэш-памяти, остальные блоки относятся к кэш-контроллеру. Адрес требуемого слова, поступающий от центрального процессора (ЦП), вводится в блок обработки адресов, в котором реализуются принятые в данной кэш-памяти принципы использования адресов при организации их сравнения с адресными тегами. Само сравнение производится в блоке сравнения адресов (БСА), который конструктивно совмещается с накопителем, если кэш-память строится по схеме ассоциативной памяти. Назначение БСА состоит в выявлении попадания или промаха при обработке запросов от центрального процессора. Если имеет место кэш-

попадание (совпадение теговой части адреса, поступающего от центрального процессора, с адресным тегом одной из ячеек кэш-памяти), то в режиме чтения информации соответствующая строка из кэш-памяти переписывается в регистр строк. С помощью селектора из неё выделяется искомое слово, которое и направляется в центральный процессор.

В случае промаха с помощью блока формирования запросов осуществляется инициализация выборки из ОП необходимой строки.

Адресация ОП при этом производится в соответствии с информацией, поступившей от центрального процессора. Выбираемая из памяти строка вместе со своим адресным тегом помещается в накопитель и регистр строк, а затем искомое слово передается в центральный процессор.

В режиме записи информации в память адрес обрабатывается также, как и при чтении. Само же слово информации из ЦП проходит через демультиплексор и заносится в регистр строк. Далее, в зависимости от выбранного способа записи оно может загрузиться в накопитель строк кэш-памяти и в ОП или только в кэш-память.

Для высвобождения места в кэш-памяти с целью записи выбираемой из ОП строки одна из строк удаляется. Определение удаляемой строки производится посредством блока замены строк, в котором хранится информация, необходимая для реализации принятой стратегии обновления находящихся в накопителе строк.

4.3.2. Способы размещения данных в кэш-памяти

Существует три основных способа размещения данных в кэш-памяти: прямое распределение (отображение), полностью ассоциативное распределение и частично ассоциативное распределение. Ниже подробно описан каждый способ размещения и его механизм преобразования адресов. Для того чтобы конкретизировать описание, положим, что кэш-память может содержать 128 строк, размер строки – 16 слов, а основная память может содержать 16384 строк. Для адресации основной памяти используется 18 бит.

Из них старшие 14 показывают адрес строки, а младшие 4 бит – адрес слова внутри этой строки. При одном обращении к памяти выбирается одна строка; 128 строк кэш-памяти указываются 7-разрядными адресами.

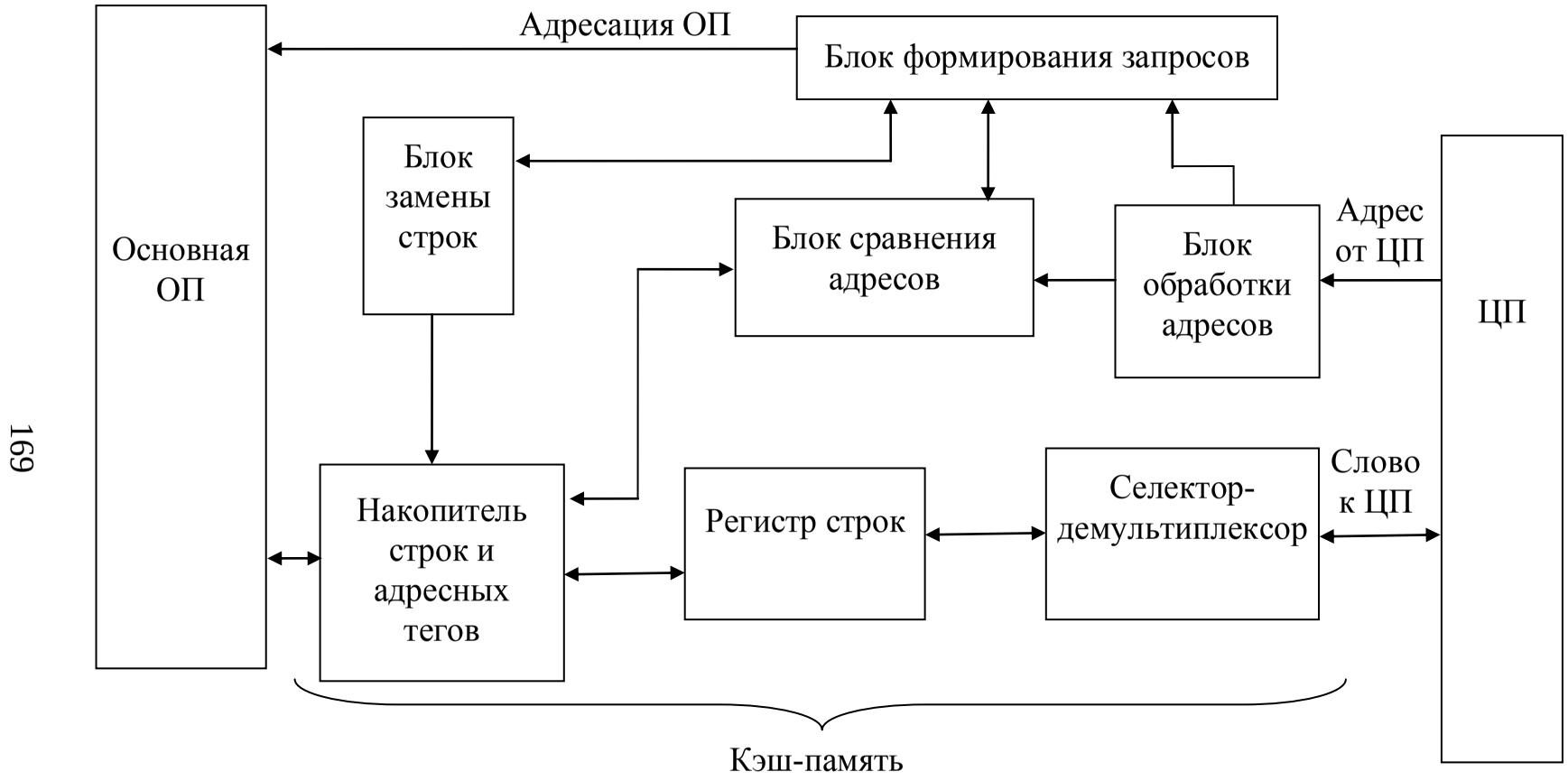


Рис. 4.3. Типовая структура кэш-памяти

Прямое распределение

При прямом распределении место хранения строк в кэш-памяти однозначно определяется по адресу строки (рис. 4.4). Адрес строки подразделяется на тег (старшие 7 бит) и индекс (младшие 7 бит).

Для того чтобы поместить в кэш-память строку из основной памяти с адресом *bn*, выбирается область внутри кэш-памяти с адресом *bm*, который равен 7 младшим битам адреса строки *bn*. Преобразование из *bn* в *bm* сводится только к выборке младших 7 бит адреса строки. По адресу *bm* в кэш-памяти может быть помещена любая из 128 строк основной памяти, имеющих адрес, 7 младших битов которого равны адресу *bm*. Для того чтобы определить, какая именно строка хранится в данное время в кэш-памяти, используется память ёмкостью 7 бит × 128 слов, в которую помещается по соответствующему адресу в качестве тега 7 старших битов адреса строки, хранящейся в данное время по адресу *bm* кэш-памяти. Это специальная память, называемая **теговой памятью.** Память, в которой хранятся строки, помещенные в кэш, называются **памятью данных**. В качестве адреса теговой памяти используются младшие 7 битов адреса строки.

При выполнении операции чтения (записи данных) из теговой памяти считывается тег. Параллельно этому осуществляется доступ к памяти данных с помощью 11 младших битов адреса основной памяти (используется 7 разрядов индекса и 4 разряда адреса слова внутри строки). Если считанный из теговой памяти тег и старшие 7 бит адреса основной памяти совпадают, то это означает, что данная строка существует в памяти данных, т.е. осуществляется кэш-попадание. В этом случае при чтении в процессор передается содержимое выбранной ячейки кэш-памяти, а при записи – в выбранную ячейку кэш-памяти загружается новая строка данных.

Если выбранный тег отличается от старших 7 бит (кэш-промах), то из основной памяти считывается соответствующая строка, а из кэш-памяти удаляется строка, определяемая 7-ю младшими разрядами адреса строки, и на его место помещается строка, считанная из основной памяти. Осуществляется также обновление соответствующего тега в теговой памяти.

Способ прямого распределения реализовать довольно просто, однако из-за того, что место хранения строки в кэш-памяти однозначно определяется по адресу строки, вероятность сосредоточения областей хранения строк в некоторой части кэш-памяти высока, т.е. замены строк будут происходить довольно часто. В такой ситуации эффективность кэш-памяти заметно снижается.

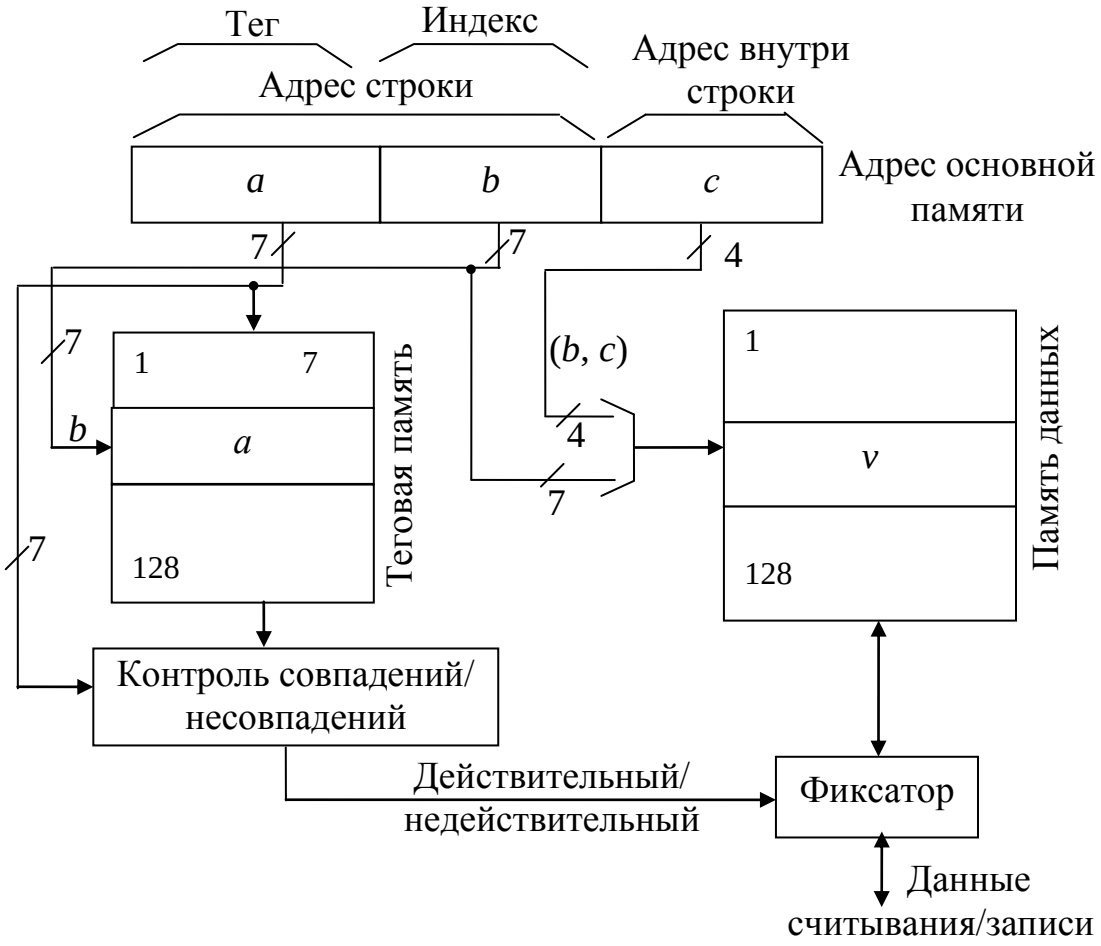


Рис. 4.4. Структура кэш-памяти с прямым распределением

Полностью ассоциативное распределение

При полностью ассоциативном распределении (fully associative) допускается размещение каждой строки основной памяти на месте любой строки кэш-памяти. Структура кэш-памяти с полностью ассоциативным распределением представлена на рис. 4.5.

Адрес основной памяти состоит из 14-разрядного адреса строки (тега) и 4-разрядного адреса внутри строки.

При полностью ассоциативном распределении механизм преобразования адресов должен быстро дать ответ, существует ли копия строки с произвольно указанным адресом в кэш-памяти, и если существует, то по какому адресу. Для этого необходимо, чтобы теговая память была реализована, как ассоциативная память. Входной информацией для ассоциативной памяти тегов (ключ поиска) является тег – 14-разрядный

адрес строки, а выходной информацией – адрес строки внутри кэш-памяти (памяти данных). Каждое слово теговой памяти состоит из 14-разрядного тега и 7-разрядного адреса строки памяти данных кэша.

Ключ поиска параллельно сравнивается со всеми тегами ассоциативной памяти. При совпадении ключа с одним из тегов теговой памяти (кэш-попадание) происходит выборка соответствующего данному тегу адреса и обращение к памяти данных.

Входной информацией для памяти данных является 11-разрядное слово (7 бит адреса строки и 4 бит адреса слова в данной строке). При выполнении операции чтения по этому адресу считывается и передается в процессор выбранная строка, а при записи – по этому же адресу в память данных записывается новая строка данных. При несовпадении ключа ни с одним из тегов теговой памяти (кэш-промах) осуществляется обращение к основной памяти и чтение необходимой строки.

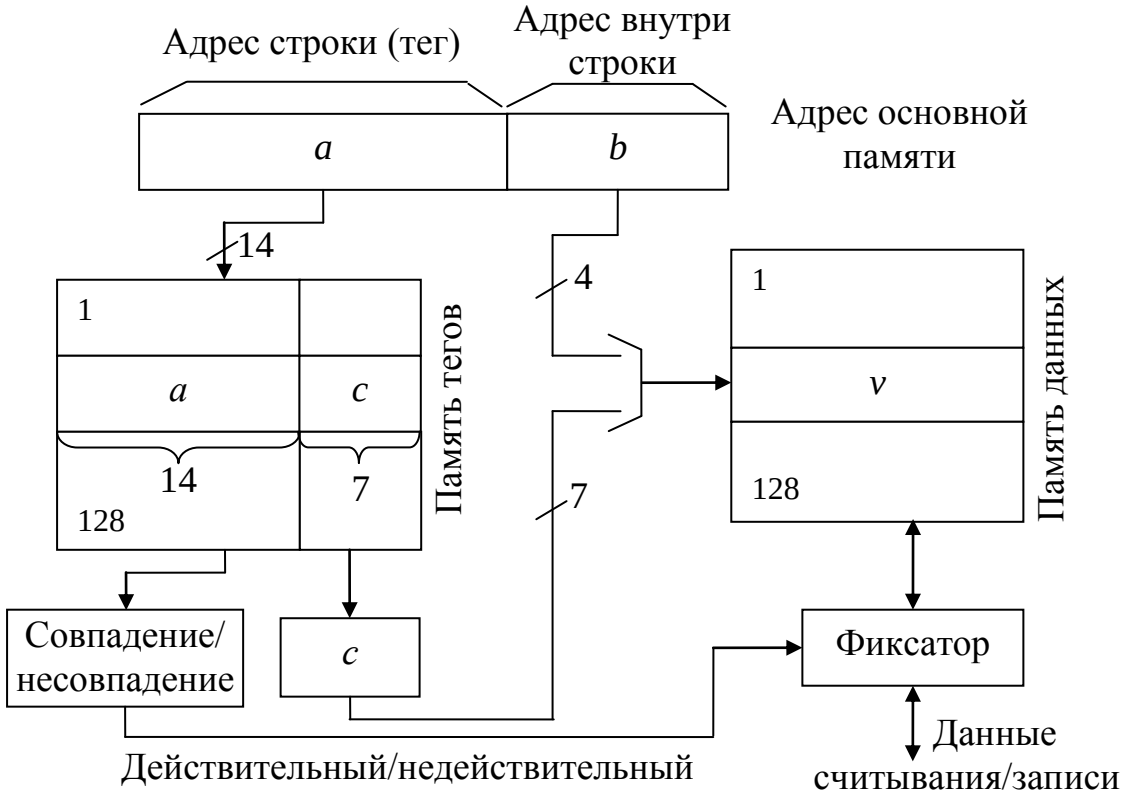


Рис. 4.5. Структура кэш-памяти с полностью ассоциативным распределением

По этому способу при замене строк кандидатом на удаление могут быть все строки в кэш-памяти.

Частично ассоциативное распределение

Если некоторая строка основной памяти может располагаться на ограниченном множестве мест в кэш-памяти, то кэш называется частично ассоциативным или множественно ассоциативным (set associative). Обычно множество представляет собой группу из двух или большего числа строк, расположенных в различных банках (блоках) данных. Если группа (множество) состоит из *n* строк (банков, блоков), то такое размещение называется частично (множественно) ассоциативным с *n* каналами (*n – way*).

В качестве примера рассмотрим структуру четырёхканальной частично ассоциативной кэш-памяти (рис. 4.6). В этом случае 4 соседних строки из 128 строк кэш-памяти образуют структуру, называемую группой.

Адрес строки основной памяти (14 бит) разделяется на две части: *b* – тег (старшие 9 бит) и *e* – адрес группы (младшие 5 бит). Адрес строки внутри кэш-памяти, состоящий из 7 бит, разделяется на адрес группы (5 бит) и адрес строки внутри группы (2 бита).

Массивы тегов и данных состоят из четырех банков данных, доступ к каждому из которых осуществляется параллельно одинаковыми адресами. Каждый банк массива тегов имеет длину слова 9 бит для помещения значения тега, а число групп тегов в банке равно 32. Каждый банк массива данных имеет длину слова такую же, как и у основной памяти, а ёмкость его определяется числом слов в одной строке, умноженным на число групп в кэш-памяти.

Для помещения в кэш-память строки, хранимой в ОП по адресу *b*, необходимо выбрать группу с адресом *e*. При этом не имеет значения, какая из четырёх строк в группе может быть выбрана. Для выбора группы используется метод прямого распределения, а для выбора строки в группе используется метод полностью ассоциативного распределения.

Когда центральный процессор запрашивает доступ по *i*-му адресу к кэш-памяти с целью чтения или записи, то осуществляется обращение к массиву тегов по адресу *e*, выбирается группа из четырёх тегов (*a, b, c, d*), каждый из которых сравнивается со старшими 9 битами (*b*) адреса строки. На выходе четырёх схем сравнения формируется унитарный код совпадения (0100), который на шифраторе преобразуется в двухразрядный позиционный код, служащий адресом для выбора банка данных (01). При операции чтения (записи) одновременно осуществляется обращение к массиву данных по адресу *e.f* (9 бит) и считывание (запись) из банка (в банк) V_2 требуемой строки или слова.

При пересылке новой строки в кэш-память удаляемая из нее строка выбирается из четырёх строк соответствующего набора (группы).

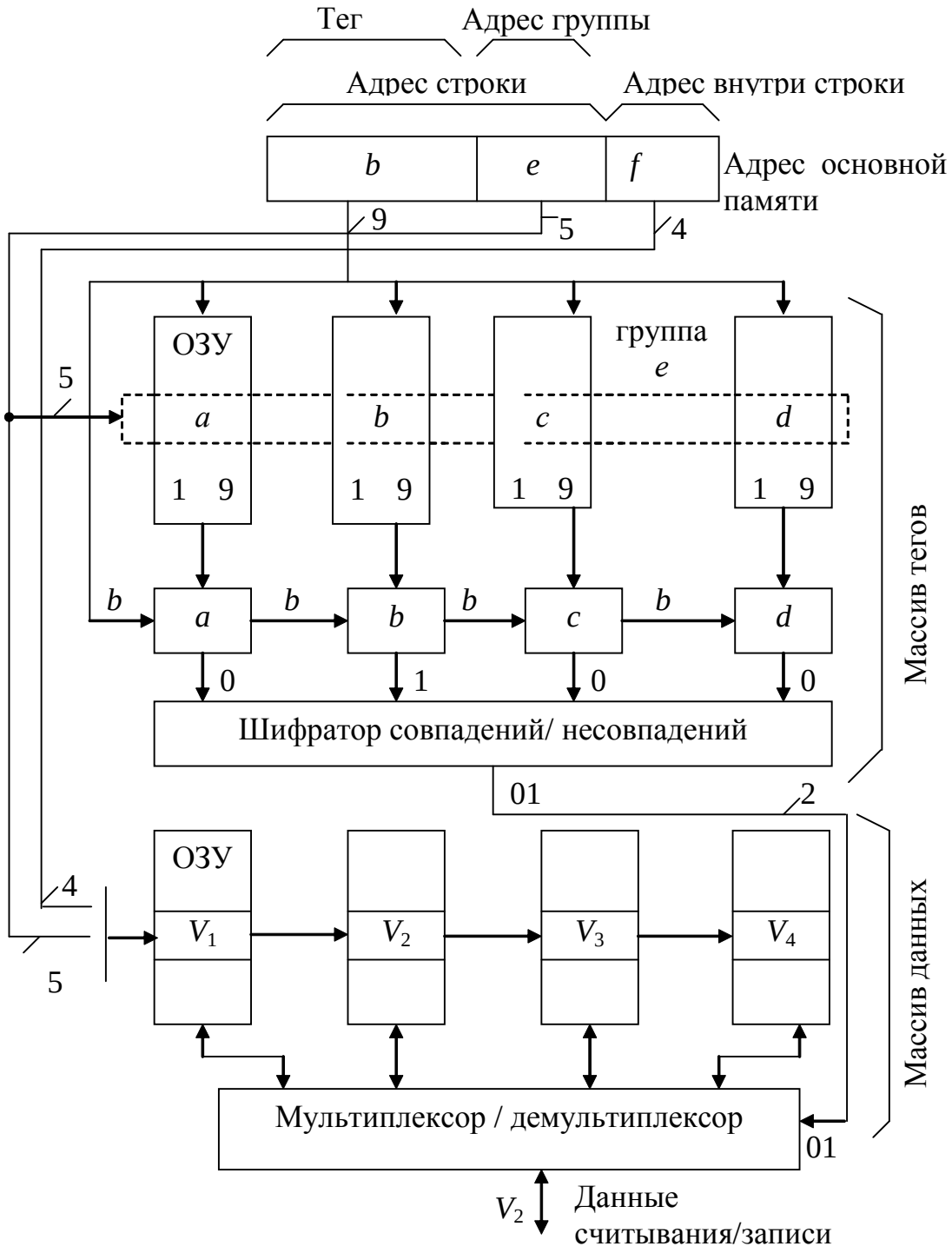


Рис. 4.6. Структура кэш-памяти с частично ассоциативным распределением

4.3.3. Методы обновления строк основной памяти и кэша

В табл. 5.1 приведены условия сохранения и обновления информации в ячейках кэш-памяти и основной памяти.

Если процессор намерен получить информацию из некоторой ячейки основной памяти, а копия содержимого этой ячейки уже имеется в кэш-памяти (первая строка табл. 4.1), то вместо оригинала считывается копия. Информация в кэш-памяти и основной памяти не изменяется. Если копии нет, то производится обращение к основной памяти. Полученная информация пересылается в процессор и попутно запоминается в кэш-памяти. Чтение информации в отсутствие копии отражено во второй строке таблицы. Информация в основной памяти не изменяется.

При записи существует несколько методов обновления старой информации. Эти методы называются **стратегией обновления строк основной памяти**. Если результат обновления строк кэш-памяти не возвращается в основную память, то содержимое основной памяти становится неадекватным вычислительному процессу. Чтобы избежать этого, предусмотрены методы обновления основной памяти, которые можно разделить на две большие группы: метод сквозной записи и метод обратной записи.

Таблица 4.1

Условия сохранения и обновления информации

Режим работы	Наличие копии ячейки ОП в кэш-памяти	Информация	
		В ячейке кэш-памяти	В ячейке основной памяти
Чтение	Копия есть. Копии нет	Не изменяется. Обновляется (создается копия)	Не изменяется. Не изменяется
Сквозная запись	Копия есть. Копии нет	Обновляется. Не изменяется	Обновляется. Обновляется
Обратная запись	Копия есть. Копии нет	Обновляется. Создается копия. Обновляется	Не изменяется. Не изменяется

Сквозная запись

По методу сквозной записи обычно обновляется слово, хранящееся в основной памяти. Если в кэш-памяти существует копия этого слова, то она также обновляется. Если же в кэш-памяти отсутствует копия этого слова, то либо из основной памяти в кэш-память пересылается строка,

содержащая это слово (метод WTWA – сквозная запись с распределением), либо этого не допускается (метод WTNWA – сквозная запись без распределения). Когда по методу сквозной записи область (строка) в кэш-памяти назначается для хранения другой строки, то в основную память можно не возвращать удаляемый блок, т.к копия там есть. Однако в этом случае эффект от использования кэш-памяти отсутствует.

Обратная запись

По методу обратной записи, если адрес объектов, по которым есть запрос обновления, существует в кэш-памяти, то обновляется только кэш-память, а основная память не обновляется. Если адреса объекта обновления нет в кэш-памяти, то в неё из основной памяти пересылается строка, содержащая этот адрес, после чего обновляется только кэш-память. По методу обратной записи, в случае замены строк, удаляемую строку необходимо также пересылать в основную память. У этого метода существуют две разновидности: метод SWB (простая обратная запись), по которому удаляемая строка возвращается в основную память, и метод FWB (флаговая обратная запись), по которому в основную память записывается только обновлённая строка кэш-памяти. В последнем случае каждая область строки в кэш-памяти снабжается однобитовым флагом, который показывает, было или нет обновление строки, хранящейся в кэш-памяти. Метод FWB обладает достаточной эффективностью, однако более эффективным считается метод FPWB (флаговая регистровая обратная запись), в котором, благодаря размещению буфера между кэш-памятью и основной памятью, предотвращается конфликт между удалением и выборкой строк.

Таким образом, теоретически более предпочтительным алгоритмом записи для кэша является метод обратной записи. Кэш с обратной записью будет хранить новую информацию до тех пор, пока у него не появится необходимость избавиться от неё. Тем самым процессор может более оперативно управлять системой. В связи с тем, что кэш со сквозной записью сразу же передаёт вновь записанную информацию в память следующего уровня, кэш со сквозной записью может вызывать дополнительные потери в быстродействии по сравнению с кэшем с обратной записью. В случае кэша с обратной записью допускается выполнение длинных последовательностей быстрых операций записи из процессора, поскольку нет необходимости немедленно направлять эти данные в основную память.

4.3.4. Методы замещения строк кэш-памяти

Способ определения строки, удаляемой из кэш-памяти, называется **стратегией замещения**. Для замещения строк кэш-памяти существует несколько методов:

- замещение строки, к которой наиболее длительное время не было обращения (метод LRU);
- замещение строки, загруженной в кэш-память первой (метод FIFO);
- произвольное замещение.

Реализация этих методов упрощается в указанной последовательности, но наибольшим эффектом обладает метод замещения наиболее давнего по использованию объекта (строки).

Для реализации этого метода необходимо манипулировать строками, которые являются объектами замещения, с помощью LRU-стека. При каждой загрузке в этот стек помещается строка, в результате чего при замене используется строка, хранящаяся в наиболее глубокой позиции стека, и эта строка удаляется из стека. При доступе к строке, которая уже содержится в LRU-стеке, эта строка удаляется из стека и заново загружается в него. Стек типа LRU устроен таким образом, что чем дольше к строке не было доступа, тем в более глубокой позиции она располагается. Реализация стека типа LRU, позволяющего с высокой скоростью выполнять такую операцию, усложняется по мере увеличения числа строк.

4.3.5. Многоуровневая организация кэша

Предельно достижимая ёмкость кэш-памяти ограничена не только её ценой, но и электромагнитной интерференцией, налагающей жёсткие ограничения на максимально возможное количество адресных линий, а значит – на непосредственно адресуемый объём памяти. В принципе, можно прибегнуть к мультиплексированию выводов или последовательной передаче адресов, но это неизбежно снизит производительность и увеличит время доступа к ячейке кэш-памяти. С другой стороны, двухпортовая статическая память действительно очень дорогая, а однопортовая не в состоянии обеспечить параллельную обработку нескольких ячеек, что приводит к досадным задержкам. Естественный выход состоит в создании многоуровневой кэш-иерархии (рис. 4.7).

Большинство современных компьютеров имеют два или три уровня кэш-памяти. Первый, наиболее «близкий» к ядру процессора (L1), обычно реализуется на быстрой двухпортовой синхронной статической

памяти, работающей на полной частоте ядра. Объём L1 кэша весьма невелик, составляет 64 Кб или 128 Кб и разделяется пополам на два кэша данных и команд для каждого ядра процессора. Латентность кэша L1 измеряется 3-мя, 4-мя тактами. На втором уровне расположен кэш L2. Он реализуется на однопортовой конвейерной статической памяти и зачастую работает на пониженной тактовой частоте. Поскольку однопортовая память значительно дешевле, объём L2 кэша достигает нескольких мегабайт в двухъядерных структурах процессоров, когда он является общим для двух ядер (Intel Core 2 Duo), или несколько сотен килобайт (256 Кб или 512 Кб), когда в многоядерном процессоре каждое ядро имеет свой L2 кэш (рис. 4.7). Этот кэш хранит как команды, так и данные. Латентность L2 для процессоров Intel Nehalem 3,2 ГГц составляет 11 тактов, для Penryn 3,2 ГГц – 18 тактов.

На третьем уровне находится L3 кэш, который объединяет ядра между собой и является разделяемым. В результате L2 кэш выступает в качестве буфера при обращениях процессорных ядер в разделяемую кэш-память, имеющую достаточно солидный объём (2 Мб – AMD K10, 8 Мб – Intel Nehalem). Латентность L3 кэша исчисляется 52-мя, 54-мя тактами.

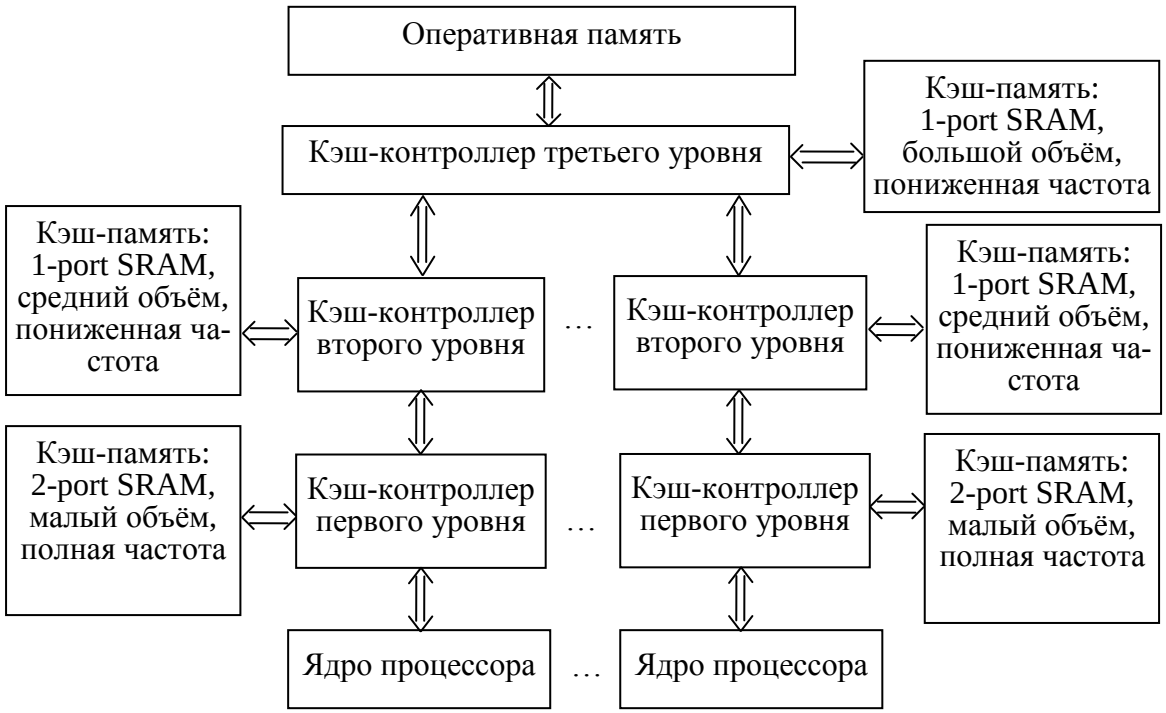


Рис. 4.7. Трехуровневая структура кэш-памяти многоядерного процессора

При построении многоуровневой кэш-памяти используют **включающую** (inclusive) или **исключающую** (exclusive) технологии. Кэш верхнего уровня, построенный по **inclusive-технологии**, всегда дублирует содержимое кэша нижнего уровня. Если построить инклюзивный L3 кэш, то он будет дублировать данные, хранящиеся в кэшах первого и второго уровней, что снижает эффективную ёмкость всей кэш-подсистемы. С другой стороны, инклюзивный разделяемый L3 кэш способен обеспечить в многоядерных процессорах более высокую скорость работы подсистемы памяти. Это связано с тем, что если ядро попытается получить доступ к данным и они отсутствуют в кэше L3, то нет необходимости искать эти данные в собственных кэшах других ядер – там их нет. А благодаря тому, что каждая строка L3 кэша снабжена дополнительными флагами, указывающими владельцев (ядра) этих данных, не вызывает затруднений и процедура обратного изменения содержимого строки кэша. Так, если какое-то ядро модифицирует данные в L3 кэше, изначально принадлежащие другому (или другим) ядру, то в этом случае обновляется содержимое L1 и L2 кэшей и этих ядер. Эта технология весьма эффективна для обеспечения когерентности персональных кэшей каждого ядра, поскольку она уменьшает потребность в обмене информацией между ядрами. По такой технологии организована кэш-память процессоров Intel Nehalem.

Кэш – подсистема, построенная по **exclusive-технологии**, никогда не хранит избыточных копий данных и потому эффективная ёмкость подсистемы определяется суммой ёмкостей кэш-памятей всех уровней. Кэш первого уровня никогда не уничтожает строки при нехватке места. Даже если они не были модифицированы, данные в обязательном порядке вытесняются в кэш второго уровня, помещаясь на то место, где находилась только что переданная кэшу L1 строка, т.е. кэши L1 и L2 как бы обмениваются друг с другом своими строками, а потому кэш-память используется весьма эффективно. По такой технологии организована кэш-память процессоров AMD K10.

4.4. Принципы организации оперативной памяти

4.4.1. Общие положения

Оперативная (основная) память представляет собой следующий уровень иерархии памяти. Оперативная память удовлетворяет запросы кэш-памяти и устройств ввода/вывода. Она является местом назначения для ввода и источником для вывода. Для оценки производительности

(быстродействия) основной памяти используются следующие параметры: **время доступа, длительность цикла памяти, латентность и пропускная способность.**

Как было сказано выше, **время доступа** – это время, проходящее с момента обращения к памяти до момента считывания данных. Данная величина приблизительно одинакова для всех типов динамической памяти и составляет примерно 50 нс. Время доступа актуально при случайном доступе к памяти, т.е. когда последовательно считываемые ячейки памяти принадлежат различным строкам матрицы памяти.

Если говорить о блочной передаче, то более показательной характеристикой является **время цикла**, т.е. время между двумя последовательными обращениями к ячейкам памяти. Первый цикл обращения всегда равен времени доступа, т.е. около 50 нс. Но при последующих циклах обращения в пределах одной страницы (строки матрицы) время существенно меньше и составляет 10 или 7,5 нс. Любая динамическая память характеризуется циклами доступа, записываемыми в виде цепочек типа 5–1–1–1 или 5–2–2–2 и т.д. Такая цепочка определяет количество тактов, необходимых для чтения первых четырех элементов (байт, слово, двойное слово) данных в страничном режиме доступа. Первая цифра в таком обозначении определяет время доступа, т.е. количество тактов, прошедших от начала обращения к банку памяти до появления данных на шине. Соответственно, при работе в страничном режиме следующие данные появятся на шине уже через меньшее количество тактов. Например, при цепочке 5–1–1–1 последующие данные появляются без задержек, т.е. с каждым тактовым импульсом.

Латентность памяти определяется некоторым набором значений временных задержек, происходящих в модуле памяти с момента прихода команды чтения (записи) до ее выполнения. Эти значения задержек принято называть **таймингами**. При описании памяти принято использовать четыре тайминга – *t*CL, *t*RCD, *t*RP, *t*RAS (иногда дополнительно указывается и Command rate), причем записываются они обычно в этой же последовательности в виде 4–4–4–12 (1T), где цифры указывают количество затраченных тактов синхронизации (в данном случае цифровые значения взяты произвольно).

Перед тем как расшифровать аббревиатуры указанных таймингов, несколько слов о принципах организации и работы оперативной памяти. Ядро памяти организовано в виде двумерной матрицы. Для получения доступа к той или иной ячейке необходимо указать адреса соответствующей строки и столбца. Для ввода адреса строки используется стробирующий сигнал RAS, а для адреса столбца – стробирующий сигнал

CAS. Порядок обращения к памяти начинается с установки регистров управления, после чего вырабатывается сигнал выбора нужного банка памяти и по прошествии (задержки) Command rate осуществляется ввод адреса строки и подача стробирующего сигнала RAS (обычно эта задержка составляет один или два такта). С приходом положительного фронта тактового импульса открывается доступ к нужной строке, а адрес строки помещается в адресный буфер строки, где он может удерживаться столько времени, сколько нужно. Через промежуток времени, называемый RAS to CAS delay (*t*RCD), т.е. задержка подачи сигнала CAS относительно сигнала RAS, подается стробирующий импульс CAS, под действием которого происходит выборка адреса столбца и открывается доступ к нужному столбцу матрицы памяти. Затем, через время CAS latency (tCL), на шине данных появляется первое слово, которое может быть считано процессором. После завершения работы со всеми ячейками активной строки выполняется команда деактивации Precharge, позволяющая перейти к следующей строке (*t*RP – это time of Row Precharge: тайминг между завершением обработки одной строки и переходом к другой). Значение *t*RAS (time of Active to Precharge Delay) считается одним из основных параметров, поскольку он описывает время задержки между моментом активации строки и моментом подачи команды деактивации Precharge, которой заканчивается работа с этой строкой. Общее правило гласит: чем меньше тайминги при одной тактовой частоте, тем быстрее память. Более того, в целом ряде случаев быстрее оказывается память с меньшими таймингами, работающая даже на более низкой тактовой частоте. Память с более высокой тактовой частотой имеет, как правило, более высокие тайминги.

Другой важнейшей характеристикой ОП является ее **пропускная способность**, которая определяется как произведение частоты работы памяти на объем данных, передаваемых за один такт. Самый простой способ увеличения максимальной пропускной способности памяти заключается в увеличении частоты ее работы. Однако на практике реализовать это совсем не просто. Вспомним, что элементарной ячейкой динамической памяти является конденсатор – инерционное по своей природе устройство. Чтобы произвести считывание информации с конденсатора, необходимо его разрядить, для чего требуется определенное время, пропорциональное емкости конденсатора, – сделать это мгновенно невозможно. Следовательно, нельзя повышать частоту ядра памяти до бесконечности. Кроме того, динамическая память требует периодической регенерации, чтобы восстанавливать заряды конденсаторов, а для зарядки конденсаторов тоже необходим определенный временной интервал. В результате повышение частоты ядра памяти сопряжено

с непреодолимыми трудностями. Конечно, применение более миниатюрных конденсаторов повышает их быстродействие, однако для этого нужно использовать иную проектную норму при производстве чипов памяти. К тому же переход на новый технологический процесс производства не может кардинально увеличить скорость работы памяти. Поэтому, кроме банального увеличения частоты работы памяти, для увеличения ее пропускной способности часто используют другие приемы.

4.4.2. Методы повышения пропускной способности ОП

Согласование производительности современных процессоров со скоростью ОП остается одной из важнейших проблем. Методы повышения производительности за счет увеличения размеров кэш-памяти и введения многоуровневой организации кэш-памяти полностью не решают эту проблему. Поэтому важным направлением современных разработок являются методы повышения пропускной способности памяти за счет ее организации, включая специальные способы организации DRAM.

Развитие способов организации памяти DDR SDRAM

Кардинальным способом увеличения пропускной способности ОП стал переход к стандарту DDR. Динамическая память DDR SDRAM пришла на смену синхронной SDRAM и обеспечила в два раза большую пропускную способность. Аббревиатура DDR (Double Data Rate) означает удвоенную скорость передачи данных. Как уже отмечалось выше, основным сдерживающим элементом увеличения тактовой частоты работы памяти является ядро памяти (массив элементов хранения – Memory Cell Array). Однако, кроме ядра, в модуле памяти присутствуют и буферы промежуточного хранения (буферы ввода/вывода – I/O Buffers), через которые ядро памяти обменивается данными с шиной памяти. Эти буферы могут иметь значительно более высокое быстродействие, чем само ядро, поэтому тактовую частоту работы шины памяти и буферов обмена можно легко увеличить. Именно такой способ и используется в DDR-памяти.

Рассмотрим предельно упрощенную схему функционирования памяти типа SDRAM (рис. 4.8, *а*). Ядро SDRAM-памяти и буферы ввода/вывода работают в синхронном режиме на одной и той же частоте. Передача каждого бита из буфера на шину происходит с каждым тактом работы ядра памяти.

При переходе от SDRAM к DDR (рис. 4.8, *б*) технология одинарной скорости передачи данных заменяется на удвоенную за счет того, что передача данных от микросхем памяти модуля к контроллеру памяти по внешней шине данных осуществляется по обоим полупериодам синхросигнала (восходящему – «фронту», и нисходящему – «срезу»). В этом и заключается суть технологии «Double Data Rate – DDR», именно поэтому «эффективная» частота памяти DDR-400 составляет 400 МГц, тогда как ее истинная частота, или частота буферов ввода/вывода, составляет 200 МГц. Таким образом, каждый буфер ввода-вывода передает на шину два бита информации за один такт, оставаясь при этом полностью синхронизированным с ядром памяти. Однако, чтобы такой режим работы стал возможным, необходимо, чтобы эти два бита были доступны буферу ввода/вывода на каждом такте работы памяти. Для этого требуется, чтобы каждая команда чтения приводила к передаче из ядра памяти в буфер сразу двух бит по двум независимым линиям передачи внутренней шины данных. Из буфера ввода/вывода биты данных затем поступают на внешнюю шину в требуемом порядке. Иными словами, можно сказать, что, при прочих равных условиях, внутренняя шина данных должна быть вдвое шире по сравнению с внешней шиной данных. Такая схема доступа к данным называется схемой «2n-предвыборки» (2n-prefetch). DDR-память, как и SDRAM, предназначалась для работы с системными частотами 100, 133, 166, 200, 216, 250 и 266 МГц. Нетрудно рассчитать пропускную способность DDR-памяти. Принимая, что ширина внешней шины данных составляет 8 байт, для памяти

DDR-400 получаем 400 МГц × 8 байт = 3,2 Гбайт/с.

Наиболее естественным путем решения проблемы достижения более высоких тактовых частот при переходе от DDR к DDR2 явилось снижение тактовой частоты внутренней шины данных вдвое по отношению к реальной тактовой частоте внешней шины данных (частоте буферов ввода/вывода). Так, в рассматриваемом примере микросхем памяти DDR2-800 (рис. 4.8, *в*) частота буферов составляет 400 МГц, а «эффективная» частота внешней шины данных – 800 МГц (поскольку сущность технологии Double Data Rate остается в силе). При этом частота внутренней шины данных (ядра памяти) составляет всего 200 МГц, поэтому для передачи 1 бита (по каждой линии данных) за такт внешней шины с «эффективной» частотой 800 МГц на каждом такте внутренней шины данных требуется передача уже 4 бит данных. Иными словами, внутренняя шина данных микросхемы памяти DDR2 должна быть в 4 раза шире по сравнению с её внешней шиной. Такая схема до-

ступа к данным называется схемой «4n-предвыборки» (4n-prefetch). Ее преимущества перед схемой 2n-prefetch, реализованной в DDR, очевидны. С одной стороны, для достижения равной пиковой пропускной способности можно использовать вдвое меньшую внутреннюю частоту микросхем памяти (200 МГц для DDR-400 и всего 100 МГц для DDR2-400), что позволяет значительно снизить энергопотребление. С другой стороны, при равной внутренней частоте функционирования микросхем DDR и DDR2 (200 МГц как для DDR-400, так и DDR2-800) последние будут характеризоваться вдвое большей теоретической пропускной способностью. Но очевидны и недостатки: функционирование буферов ввода/вывода микросхем DDR2 на вдвое большей частоте и использование более сложной схемы преобразования «4–1» приводит к ощутимому возрастанию задержек (таймингов).

Очередной «эволюционный скачок» в технологии реализации памяти DDR SDRAM – это переход от стандарта DDR2 к новому стандарту DDR3. Нетрудно догадаться, что основной принцип, лежащий в основе перехода от DDR2 к DDR3, в точности повторяет рассмотренную выше идею, заложенную при переходе от DDR к DDR2.

А именно, DDR3 – это всё та же удвоенная частота внешней шины данных по отношению к частоте внутренней шины, это удвоенная частота буферов ввода/вывода по сравнению с DDR2. Типичными скоростными категориями памяти нового стандарта DDR3 являются разновидности DDR3-800, DDR3-1066, DDR3-1333, DDR3-1600, DDR3-1866. Очередное увеличение теоретической пропускной способности компонентов памяти в 2 раза вновь связано со снижением их внутренней частоты функционирования во столько же раз. Поэтому для достижения темпа передачи данных со скоростью 1 бит/такт по каждой линии внешней шины данных с «эффективной» частотой в 1600 МГц (как в примере, рассмотренном на рис. 4.8, *г*) используемые микросхемы (с частотой 200 МГц) должны передавать по 8 бит данных за каждый «свой» такт, т.е. ширина внутренней шины данных микросхем памяти окажется уже в 8 раз больше по сравнению с шириной их внешней шины. Такая схема передачи данных с рассмотренным преобразованием типа «8–1» называется схемой «8n-предвыборки» (8n-prefetch).

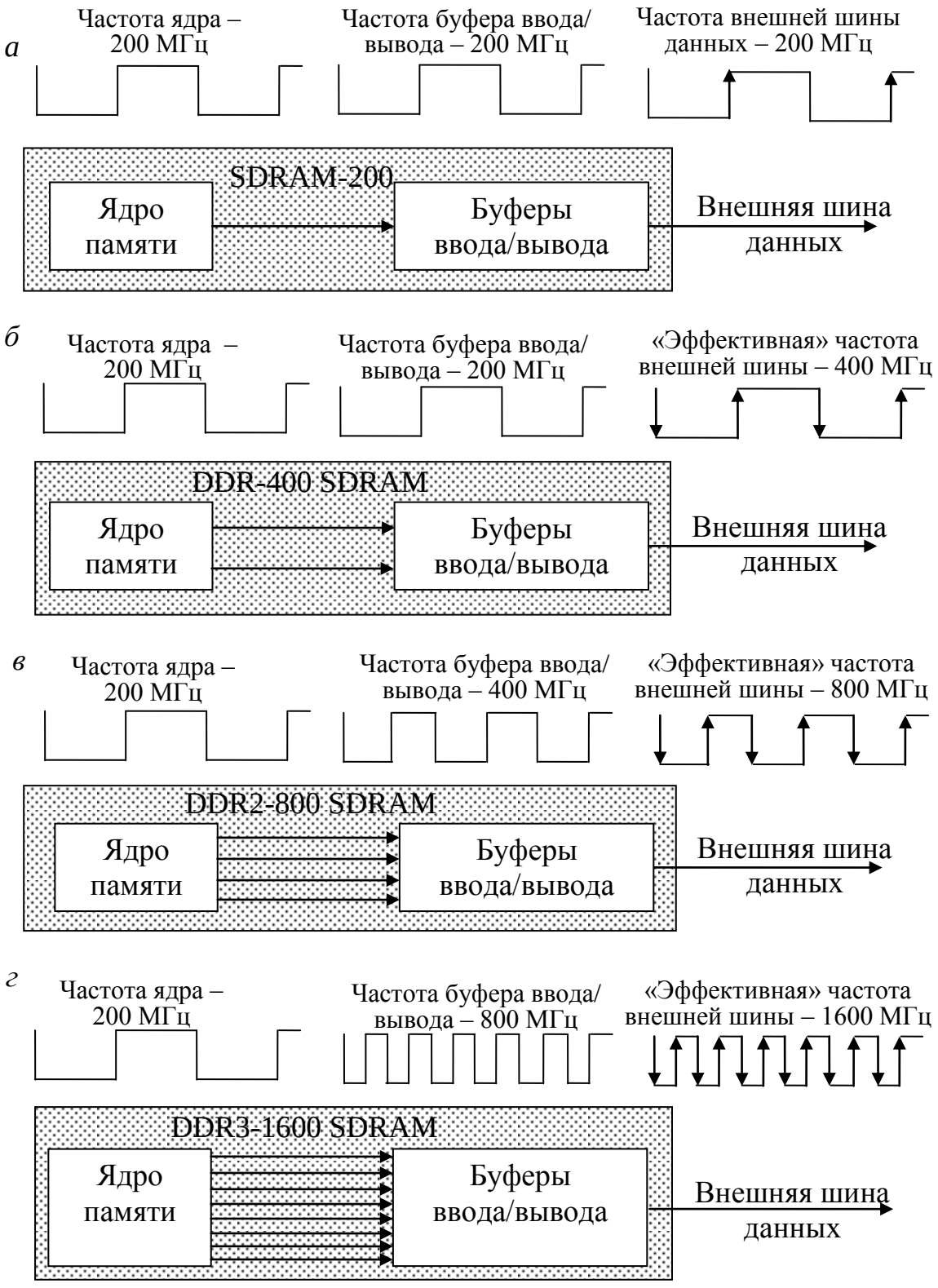


Рис. 4.8. Схематическое представление передачи данных в микросхеме памяти: *а* – SDRAM-200; *б* – DDR-400; *в* – DDR2-800; *г* – DDR3-1600

Преимущества при переходе от DDR2 к DDR3 те же, что и при состоявшемся ранее переходе от DDR к DDR2: с одной стороны, это снижение энергопотребления компонентов в условиях равенства их пиковой пропускной способности (DDR3-800 против DDR2-800), с другой стороны – возможность дальнейшего наращивания тактовой частоты и теоретической пропускной способности при сохранении прежнего уровня «внутренней» частоты компонентов (DDR3-1600 против DDR2-800). Теми же являются и недостатки: дальнейший разрыв между «внутренней» и «внешней» частотами шин компонентов памяти приводит к ещё большим задержкам.

Дальнейшее развитие технологии реализации памяти DDR SDRAM – это переход к новому стандарту DDR4. По сравнению с DDR3 эта память работает на более высоких частотах (DDR4 – 3200, DDR4 – 3000, DDR4 – 2800, DDR4 – 2666), благодаря чему она может обеспечить гораздо лучшую пропускную способность, однако при этом её латентность заметно выше.

Чтобы проиллюстрировать вышесказанное приведем (рис. 4.9) с помощью бенчмарков из пакета SiSoftware Sandra 21.42 практические характеристики подсистемы памяти Skylake-S, укомплектованной модулями DDR4 SDRAM с различной скоростью. Для сравнения рядом приводятся результаты, полученные в аналогичной системе с процессором Haswell и привычной памятью типа DDR3 SDRAM.

Как нетрудно заметить, DDR4 SDRAM действительно обеспечивает более высокую пропускную способность. Она продолжает масштабироваться синхронно с частотой, то есть модули DDR4 с более высокой задекларированной скоростью всегда смогут обеспечить лучшую полосу пропускания по сравнению с DDR3. И это – несомненное преимущество новой технологии. Однако латентность серьёзно повысилась. Как видно из диаграмм, такие же, как у DDR3-1866, задержки можно получить лишь при установке в систему модулей класса DDR4-3000, которые относятся к числу премиальных оверклокерских предложений. Иными словами, переход с DDR3 на DDR4 пока не несёт очевидных плюсов. В каких-то аспектах быстродействия системы нового поколения от использования иной технологии памяти выиграют, но в каких-то и проиграют.

Правда, не стоит упускать из виду, что с выходом Skylake внедрение DDR4 в массовых системах должно заметно ускорить продвижение этой технологии. Так что недорогие и скоростные модули DDR4 SDRAM, которые будут превосходить память предшествующего стан-

дарта во всех аспектах, могут приобрести широкое распространение уже в самом ближайшем будущем.

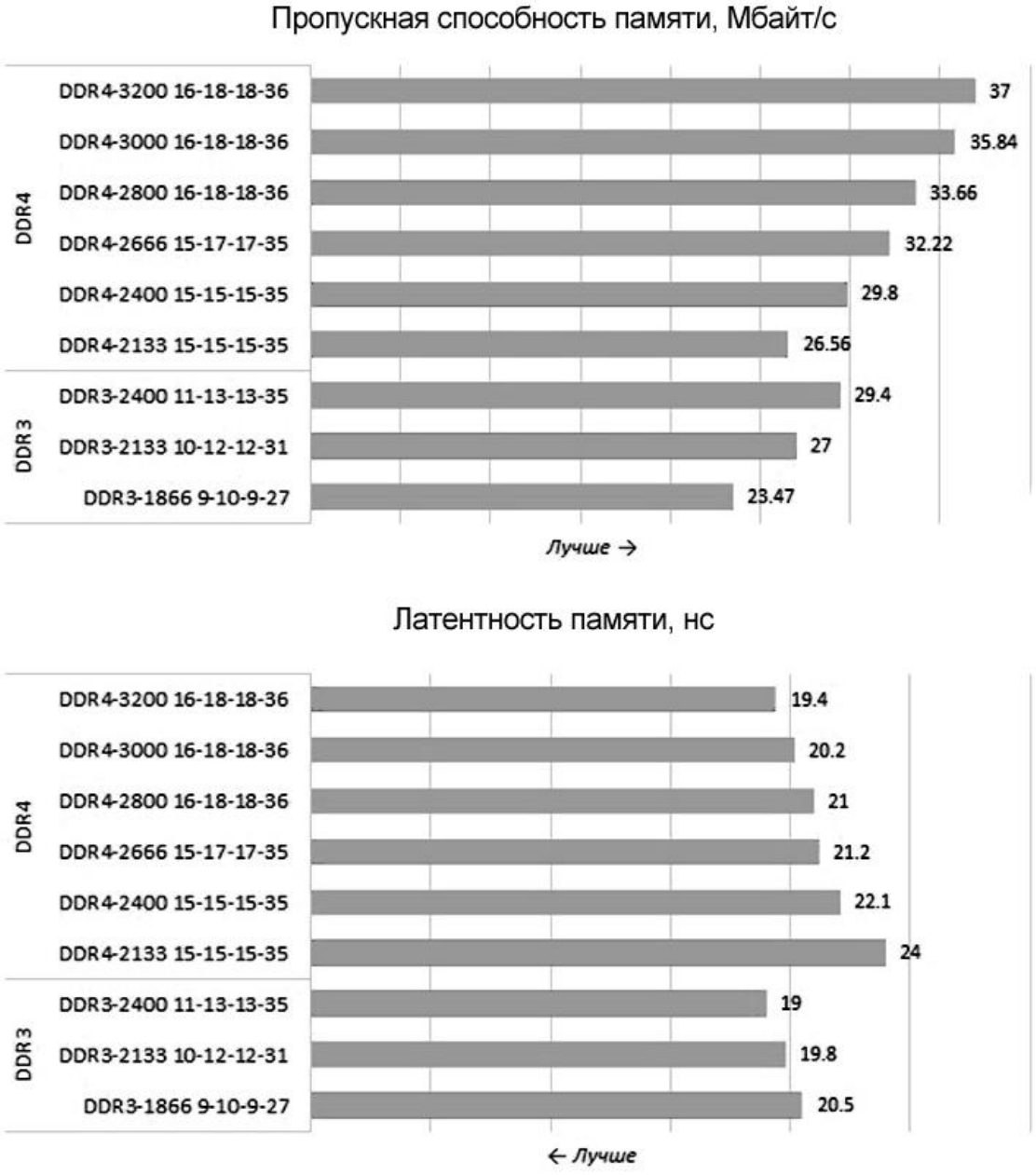


Рис. 4.9. Характеристики памяти на DDR4 в сравнении с DDR3

Выборка широким словом

Прямой способ сокращения числа обращений к ОП состоит в организации **выборки широким словом.** Этот способ основывается на свойстве локальности данных и программ. При выборке широким словом за одно обращение к ОП производится одновременная запись или считывание нескольких команд или слов данных из «широкой» ячейки. Широкое слово заносится в буферную память (кэш-память) или регистр, где оно расформировывается на отдельные команды или слова данных, которые могут последовательно использоваться процессором без дополнительных обращений к ОП.

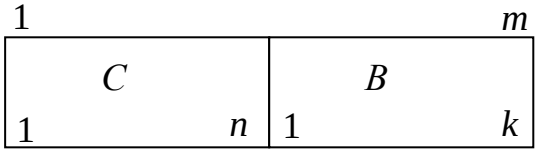
В системах с кэш-памятью первого уровня ширина шин данных ОП часто соответствует ширине шин данных кэш-памяти, которая во многих случаях имеет физическую ширину шин данных, соответствующую количеству разрядов в слове. Удвоение и учетверение ширины шин кэш-памяти и ОП удваивает или учетверяет соответственно полосу пропускания системы памяти.

Реализация выборки широким словом вызывает необходимость мультиплексирования данных между кэш-памятью и процессором, поскольку основной единицей обработки данных в процессоре все еще остается слово. Эти мультиплексоры оказываются на критическом пути поступления информации в процессор. Кэш-память второго уровня несколько смягчает эту проблему, в этом случае мультиплексоры могут располагаться между двумя уровнями кэш-памяти, т.е. вносимая ими задержка не столь критична. Другая проблема, связанная с увеличением разрядности памяти, заключается в необходимости определения минимального объема (инкремента) памяти для поэтапного её расширения, которое часто выполняется самими пользователями во время эксплуатации системы. Удвоение или учетверение ширины памяти приводит к удвоению или учетверению этого минимального инкремента. Кроме того, имеются проблемы и с организацией коррекции ошибок в системах с широкой памятью.

Расслоение обращений

Другой способ повышения пропускной способности ОП связан с построением памяти, состоящей на физическом уровне из нескольких модулей (банков) с автономными схемами адресации, записи и чтения. При этом на логическом уровне управления памятью организуются последовательные обращения к различным физическим модулям. Обращения к различным модулям могут перекрываться, и таким образом об-

разуется своеобразный конвейер. Эта процедура носит название **расслоения памяти.** Целью данного метода является увеличение скорости доступа к памяти посредством совмещения фаз обращений ко многим модулям памяти. Известно несколько вариантов организации расслоения. Наиболее часто используется способ расслоения обращений за счет **расслоения адресов.** Этот способ основывается на свойстве локальности программ и данных, предполагающем, что адрес следующей команды программы на единицу больше адреса предыдущей (линейность программ нарушается только командами перехода). Аналогичная последовательность адресов генерируется процессором при чтении и записи слов данных. Таким образом, типичным случаем распределения адресов обращений к памяти является последовательность вида a, $a + 1$, $a + 2$, … Из этого следует, что расслоение обращений возможно, если ячейки с адресами a, $a + 1$, $a + 2$, … будут размещаться в блоках 0, 1, 2, … Такое распределение ячеек по модулям (банкам) обеспечивается за счет использования адресов вида



,

где B – k-разрядный адрес модуля (младшая часть адреса) и C – n-разрядный адрес ячейки в модуле B (старшая часть адреса).

Принцип расслоения обращений иллюстрируется на рис. 4.9, *а*. Все программы и данные «размещаются» в адресном пространстве последовательно. Однако ячейки памяти, имеющие смежные адреса, находятся в различных физических модулях памяти. Если ОП состоит из 4-х модулей, то номер модуля кодируется двумя младшими разрядами адреса. При этом полные m-разрядные адреса 0, 4, 8, … будут относиться к блоку 0, адреса 1, 5, 9, … – к блоку 1, адреса 2, 6, 10, … – к блоку 2 и адреса 3, 7, 11, … – к блоку 3. В результате этого последовательность обращений к адресам 0, 1, 2, 3, 4, 5, … будет расслоена между модулями 0, 1, 2, 3, 0, 1, ….

Поскольку каждый физический модуль памяти имеет собственные схемы управления выборкой, можно обращение к следующему модулю производить, не дожидаясь ответа от предыдущего. Так на временной диаграмме (рис. 4.9, *б*) показано, что время доступа к каждому модулю составляет $\tau = 4T$, где $T = t_{i+1} - t_i$ – длительность такта. В каждом такте

следуют непрерывно обращения к модулям памяти в моменты времени t_1, t_2, t_3, ...

При наличии четырех модулей темп выдачи квантов информации из памяти в процессор будет соответствовать одному такту T, при этом скорость выдачи информации из каждого модуля в четыре раза ниже.

Задержка в выдаче кванта информации относительно момента обращения также составляет $4T$, однако задержка в выдаче каждого последующего кванта относительно момента выдачи предыдущего составит T.

При реализации расслоения по адресам число модулей памяти может быть произвольным и необязательно кратным степени 2. В некоторых компьютерах допускается произвольное отключение модулей памяти, что позволяет исключать из конфигурации неисправные модули.

В современных высокопроизводительных компьютерах число модулей обычно составляет 4–16, но иногда превышает 64.

Так как схема расслоения по адресам базируется на допущении о локальности, она дает эффект в тех случаях, когда это допущение справедливо, т.е. при решении одной задачи.

Для повышения производительности мультипроцессорных систем, работающих в многозадачных режимах, реализуют другие схемы, при которых **различные процессоры обращаются к различным модулям** памяти. Необходимо помнить, что процессоры ввода/вывода также занимают циклы памяти и вследствие этого могут сильно влиять на производительность системы. Для уменьшения этого влияния обращения центрального процессора и процессоров ввода/вывода можно организовать к разным модулям памяти.

Обобщением идеи расслоения памяти является **возможность реализации нескольких независимых обращений**, когда несколько контроллеров памяти позволяют модулям памяти (или группам расслоенных модулей памяти) работать независимо.

Прямое уменьшение числа конфликтов за счет организации чередующихся обращений к различным модулям памяти может быть достигнуто путем **размещения программ и данных в разных модулях.** Доля команд в программе, требующих ссылок к находящимся в ОП данным, зависит от класса решаемой задачи и от архитектурных особенностей компьютера. Для большинства ЭВМ с традиционной архитектурой и задач научно-технического характера эта доля превышает 50 %.

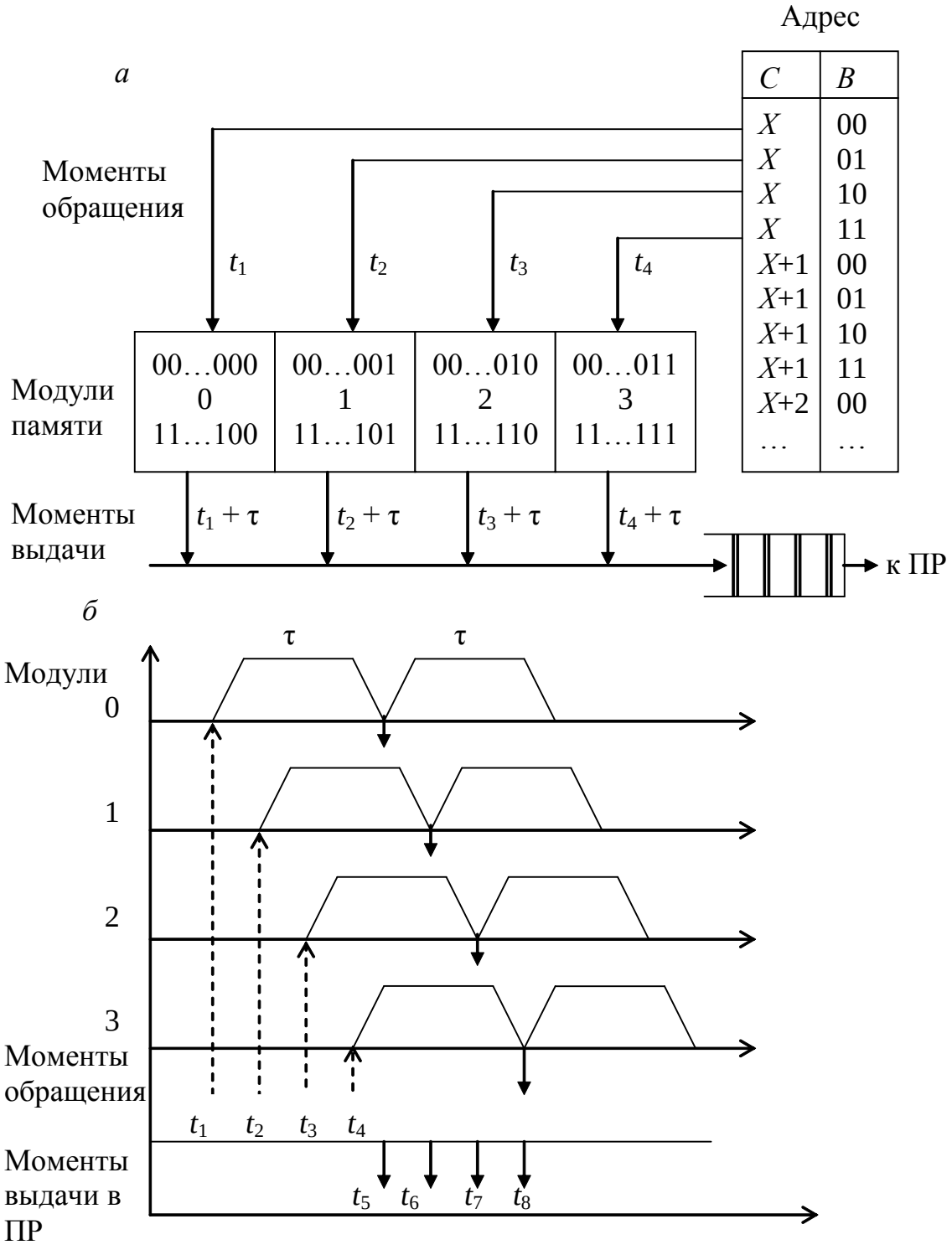


Рис. 4.9. Расслоение памяти:
а – организация адресного пространства; *б* – временная диаграмма работы модулей

Поскольку обращения к командам и элементам данных чередуются, то разделение памяти на память команд и память данных повышает быстродействие машины подобно рассмотренному выше механизму расслоения. Разделение памяти на память команд и память данных широко используется в системах управления или обработки сигналов. В подобного рода системах в качестве памяти команд нередко используются постоянные запоминающие устройства (ПЗУ), цикл которых меньше цикла устройств, допускающих запись, это делает разделение программ и данных весьма эффективным. Следует отметить, что обращения процессоров ввода/вывода в режиме прямого доступа в память логически реализуются как обращения к памяти данных.

Выбор той или иной схемы расслоения для компьютера (системы) определяется целями (достижение высокой производительности при решении множества задач или высокого быстродействия при решении одной задачи), архитектурными и структурными особенностями системы, а также элементной базой (соотношением длительностей циклов памяти и узлов обработки). Могут использоваться комбинированные схемы расслоения.

4.4.3. Методы управления памятью

Оперативная память является важнейшим и наиболее дефицитным ресурсом в вычислительных машинах и системах, требующим тщательного и эффективного управления. Проблема усложняется при переходе к мультипрограммным системам, т.к в них оперативную память одновременно используют несколько вычислительных процессов (программ).

Типы адресов

Для идентификации переменных и команд используются символьные имена (метки), виртуальные адреса и физические адреса (рис. 4.10).

Символьные имена присваивает пользователь при написании программы на алгоритмическом языке или ассемблере.

Виртуальные адреса вырабатывает транслятор, переводящий программу на машинный язык. Так как во время трансляции в общем случае неизвестно, в какое место ОП будет загружена программа, то транслятор присваивает переменным и командам виртуальные (условные) адреса, обычно считая по умолчанию, что программа будет размещена, начиная с нулевого адреса. Совокупность виртуальных адресов процесса (программы) называется виртуальным адресным пространством. Каждый процесс имеет собственное виртуальное адресное простран-

ство. Максимальный размер виртуального адресного пространства ограничивается разрядностью адреса, присущей данной архитектуре компьютера и, как правило, не совпадает с объемом физической памяти, имеющимся в компьютере.

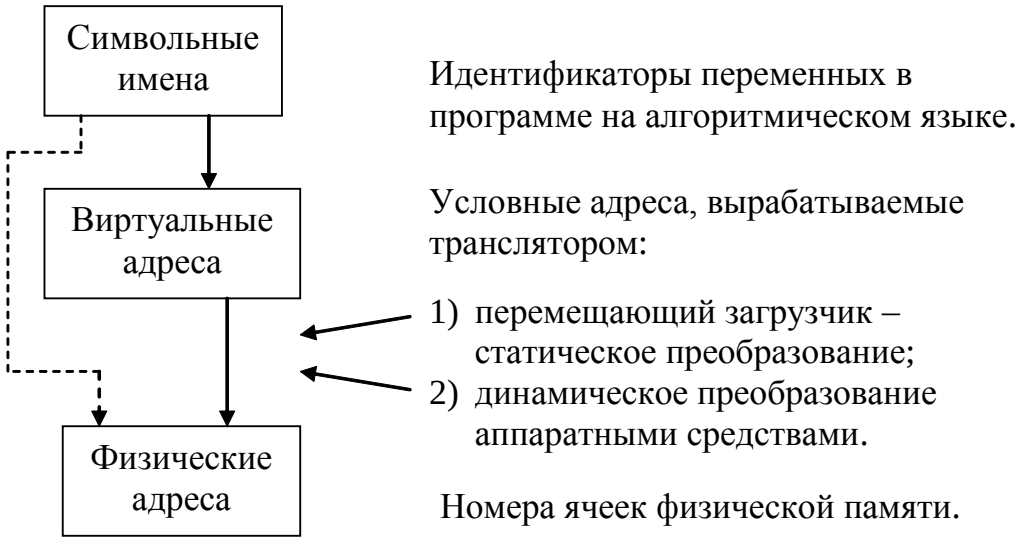


Рис. 4.10. Типы адресов

Физические адреса соответствуют номерам ячеек ОП, где в действительности расположены или будут расположены переменные и команды. Переход от виртуальных адресов к физическим может осуществляться двумя способами.

В первом случае замену виртуальных адресов на физические делает специальная системная программа – **перемещающий загрузчик**. Перемещающий загрузчик на основании имеющихся у него исходных данных о начальном адресе физической памяти, в которую предстоит загружать программу, и информации, предоставленной транслятором об адресно-зависимых константах программы, выполняет загрузку программы, совмещая её с заменой виртуальных адресов физическими.

Второй способ заключается в том, что программа загружается в память в неизменном виде в виртуальных адресах, при этом операционная система (ОС) фиксирует смещение действительного расположения программного кода относительно виртуального адресного пространства. Во время выполнения программы при каждом обращении к ОП выполняется **преобразование виртуального адреса в физический**. Второй способ является более гибким, он допускает перемещение программы во время ее выполнения, в то время как перемещающий загруз-

чик жестко привязывает программу к первоначально выделенному ей участку. Вместе с тем использование загрузчика уменьшает накладные расходы, т.к преобразование каждого виртуального адреса происходит только один раз во время загрузки, а во втором случае – каждый раз при обращении по данному адресу.

В некоторых случаях (обычно в специализированных системах), когда заранее точно известно, в какой области ОП будет выполняться программа, транслятор выдает исполняемый код сразу в физических адресах.

Классификация методов распределения оперативной памяти

Все методы управления памятью могут быть разделены на два класса (рис. 4.11):

- методы распределения ОП без использования внешней памяти (дискового пространства);
- методы распределения памяти с использованием дискового пространства.

Первая группа включает методы распределения памяти фиксированными, динамическими и перемещаемыми разделами. Вторая – страничное, сегментное и странично-сегментное распределение памяти.

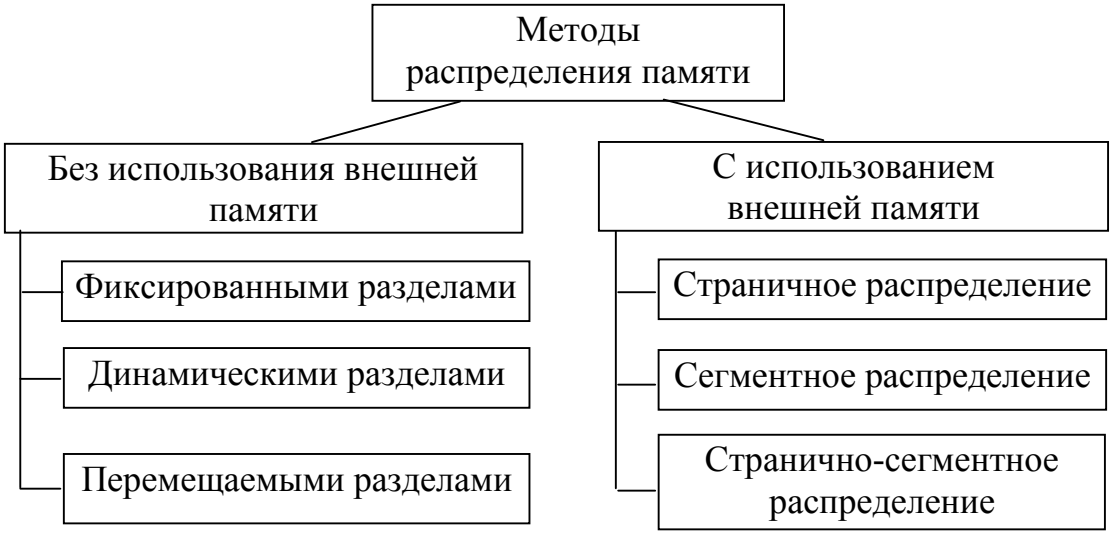


Рис. 4.11. Классификация методов распределения ОП

Рассмотрим вначале первую группу методов.

Распределение памяти фиксированными разделами

Самым простым способом управления оперативной памятью является разделение её на несколько разделов (сегментов) фиксированной величины (**статическое распределение**). Это может быть выполнено вручную оператором во время старта системы или во время её генерации. Очередная задача, поступающая на выполнение, помещается либо в общую очередь (рис. 4.12, *а*), либо в очередь к некоторому разделу (рис. 4.12, *б*). Подсистема управления памятью в этом случае выполняет следующие задачи: сравнивает размер программы, поступившей на выполнение, и свободных разделов памяти; выбирает подходящий раздел; осуществляет загрузку программы и настройку адресов.

При очевидном преимуществе, заключающемся в простоте реализации, данный метод имеет существенный недостаток – **жёсткость**. Так как в каждом разделе может выполняться только одна программа, то уровень мультипрограммирования заранее ограничен числом разделов независимо от того, какой размер имеют программы.

Даже если программа имеет небольшой объём, она будет занимать весь раздел, что приводит к неэффективному использованию памяти. С другой стороны, даже если объём оперативной памяти машины позволяет выполнить некоторую программу, разбиение памяти на разделы не позволяет сделать этого.

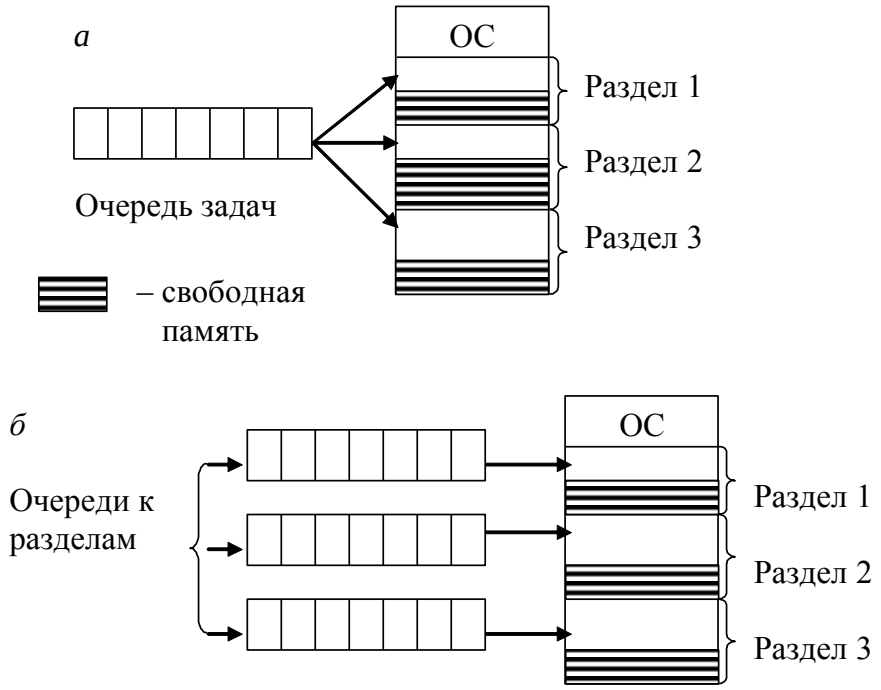


Рис. 4.12. Распределение памяти фиксированными разделами:
а – с общей очередью; *б* – с отдельными очередями

Распределение памяти разделами переменной величины

В этом случае память машины не делится заранее на разделы. Сначала вся память свободна. Каждой вновь поступающей задаче выделяется необходимая ей память. Если достаточный объём памяти отсутствует, то задача не принимается на выполнение и стоит в очереди. После завершения задачи память освобождается и на это место может быть загружена другая задача. Таким образом, в произвольный момент времени оперативная память представляет собой случайную последовательность занятых и свободных участков (разделов) произвольного размера. На рис. 4.13 показано состояние памяти в различные моменты времени при использовании **динамического распределения**. Так, в момент t_0 в памяти находится только ОС, а к моменту t_1 память разделена между 5 задачами, причём задача П4, завершая работу, покидает память к моменту t_2. На освободившееся место загружается задача П6, поступившая в момент t_3.

Задачами операционной системы при реализации данного метода управления памятью являются: ведение таблиц свободных и занятых областей, в которых указываются начальные адреса и размеры участков памяти; анализ запроса (при поступлении новой задачи); просмотр таблицы свободных областей и выбор раздела, размер которого достаточен для размещения поступившей задачи: загрузка задачи в выделенный ей раздел и корректировка таблиц свободных и занятых областей; корректировка таблиц свободных и занятых областей (после завершения задачи).

Программный код не перемещается во время выполнения, т.е. может быть проведена единовременная настройка адресов посредством использования перемещающего загрузчика.

Выбор раздела для вновь поступившей задачи может осуществляться по разным правилам: «первый попавшийся раздел достаточного размера»; «раздел, имеющий наименьший достаточный размер»; «раздел, имеющий наибольший достаточный размер». Все эти правила имеют свои преимущества и недостатки.

По сравнению с методом распределения памяти фиксированными разделами данный метод обладает гораздо большей гибкостью, но ему присущ очень серьезный недостаток – **фрагментация памяти**. Фрагментация – это наличие большого числа несмежных участков свободной памяти очень маленького размера (фрагментов). Настолько маленького, что ни одна из вновь поступающих программ не может поместиться ни в одном из участков, хотя суммарный объем фрагментов может составить значительную величину, намного превышающую требуемый объём памяти.

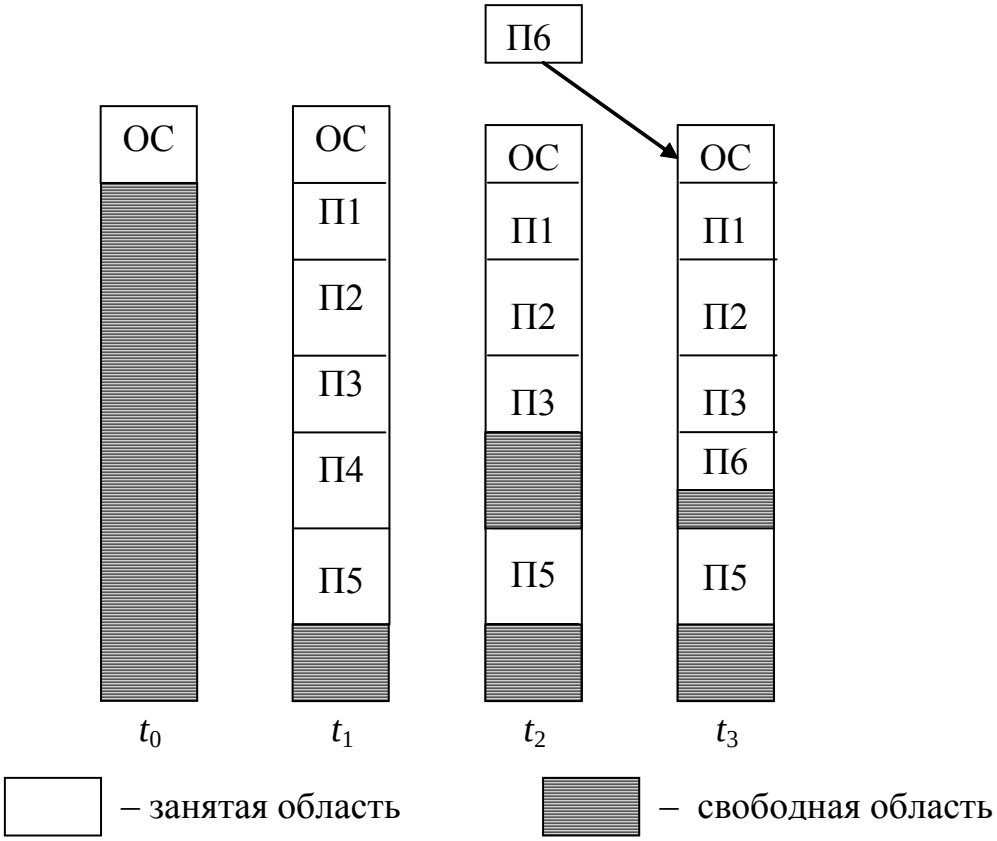


Рис. 4.13. Распределение памяти динамическими разделами

Перемещаемые разделы

Одним из методов борьбы с фрагментацией является перемещение всех занятых участков в сторону старших либо в сторону младших адресов так, чтобы вся свободная память образовывала единую свободную область (рис. 4.14). В дополнение к функциям, которые выполняет ОС при распределении памяти переменными разделами, в данном случае она должна еще время от времени копировать содержимое разделов из одного места памяти в другое, корректируя таблицы свободных и занятых областей.

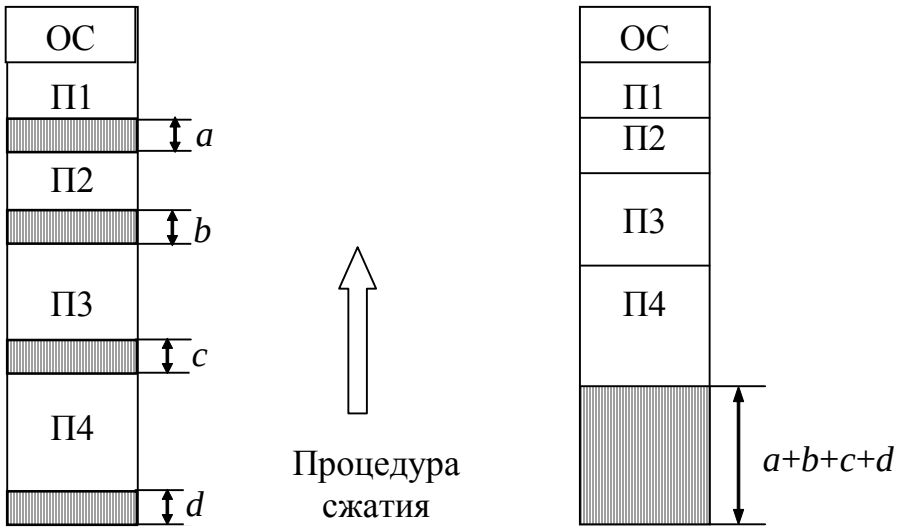


Рис. 4.14. Распределение памяти перемещаемыми разделами

Эта процедура называется **сжатием**. Сжатие может выполняться либо при каждом завершении задачи, либо только тогда, когда для вновь поступившей задачи нет свободного раздела достаточного размера. В первом случае требуется меньше вычислительной работы при корректировке таблиц, а во втором – реже выполняется процедура сжатия. Так как программы перемещаются по оперативной памяти в ходе своего выполнения, то преобразование адресов из виртуальной формы в физическую должно выполняться динамическим способом.

Хотя процедура сжатия и приводит к более эффективному использованию памяти, она может потребовать значительного времени, что часто перевешивает преимущества данного метода.

4.4.4. Организация виртуальной памяти

Концепция виртуальной памяти

Общепринятая в настоящее время концепция виртуальной памяти появилась достаточно давно. Она позволила решить целый ряд актуальных вопросов организации вычислений. Прежде всего к числу таких вопросов относится обеспечение надежного функционирования мультипрограммных систем. В любой момент времени компьютер выполняет множество процессов (или задач), каждый из которых располагает своим адресным пространством. Было бы слишком накладно отдавать всю физическую оперативную память какой-то одной задаче, тем более, что многие задачи реально используют только небольшую часть своего ад-

ресного пространства. Поэтому необходим механизм разделения небольшой физической памяти между различными задачами. Виртуальная память является одним из способов реализации такой возможности. Она делит физическую память на блоки и распределяет их между различными задачами, при этом она предусматривает также некоторую схему защиты, которая ограничивает задачу теми блоками, которые ей принадлежат. Большинство типов виртуальной памяти сокращают также время начального запуска программы на процессоре, поскольку не весь программный код и данные требуются ей в физической памяти, чтобы начать выполнение.

Другой вопрос, тесно связанный с реализацией концепции виртуальной памяти, касается организации вычислений на компьютере задач очень большого объёма. Раньше, если программа становилась слишком большой для физической оперативной памяти, часть её необходимо было хранить во внешней памяти (на диске) и задача приспособить её для решения на компьютере ложилась на программиста. Программисты делили программы на части и затем определяли те из них, которые можно было бы выполнять независимо, организуя оверлейные структуры, которые загружались в основную память и выгружались из неё под управлением программы пользователя. Программист должен был следить за тем, чтобы программа не обращалась вне отведённого ей пространства физической памяти. Виртуальная память освободила программистов от этого бремени.

Виртуальным называется такой ресурс, который для пользователя (пользовательской программы) представляется обладающим свойствами, которыми он в действительности не обладает. Так, например, пользователю может быть предоставлена виртуальная оперативная память, размер которой превосходит всю имеющуюся в системе реальную ОП. Пользователь пишет программы так, как будто в его распоряжении имеется однородная (одноуровневая) оперативная память большого объёма, но в действительности все данные, используемые программой, хранятся на нескольких разнородных запоминающих устройствах, обычно в ОП и на дисках, и при необходимости частями перемещаются между ними. Все эти действия выполняются автоматически, без участия программиста, т.е. механизм виртуальной памяти является прозрачным по отношению к пользователю.

Наиболее распространенными реализациями виртуальной памяти являются страничное, сегментное и странично-сегментное распределение памяти (рис. 4.11).

Страничное распределение

На рис. 4.15 показана схема страничного распределения памяти. Виртуальное адресное пространство каждого процесса делится на части, называемые **виртуальными страницами,** одинакового, фиксированного (для данной системы) размера. В общем случае размер виртуального адресного пространства не является кратным размеру страницы, поэтому последняя страница каждого процесса дополняется фиктивной областью.

Вся оперативная память машины также делится на части такого же размера, называемые **физическими страницами** (или блоками).

Размер страницы обычно выбирается равным степени двойки: 512, 1024 и т.д., это позволяет упростить механизм преобразования адресов.

При загрузке процесса часть его виртуальных страниц помещается в оперативную память, а остальные – на диск. Смежные виртуальные страницы необязательно располагаются в смежных физических страницах. При загрузке операционная система создает для каждого процесса информационную структуру – **таблицу страниц**, в которой устанавливается соответствие между номерами виртуальных и физических страниц для страниц, загруженных в оперативную память, или делается отметка о том, что виртуальная страница выгружена на диск (ВЗУ). Кроме того, в таблице страниц содержится управляющая информация, такая как признак модификации страницы, признак невыгружаемости (выгрузка некоторых страниц может быть запрещена), признак обращения к странице (используется для подсчёта числа обращений за определённый период времени) и другие данные, формируемые и используемые механизмом виртуальной памяти.

В данной ситуации может быть использовано много разных критериев выбора, наиболее популярные из них следующие:

- дольше всего не использовавшаяся страница;
- первая попавшаяся страница;
- страница, к которой в последнее время было меньше всего обращений.

В некоторых системах используется понятие рабочего множества страниц. Рабочее множество определяется для каждого процесса и представляет собой перечень наиболее часто используемых страниц, которые должны постоянно находиться в оперативной памяти и поэтому не подлежат выгрузке.

Рис. 4.15. Страничное распределение памяти

После того как выбрана страница, которая должна покинуть оперативную память, анализируется ее признак модификации (из таблицы страниц). Если выталкиваемая страница с момента загрузки была модифицирована, то ее новая версия должна быть переписана на диск. Если нет, то она может быть просто уничтожена, т.е. соответствующая физическая страница объявляется свободной.

Рассмотрим механизм преобразования виртуального адреса в физический при страничной организации памяти (рис. 4.16).

Рис. 4.16. Механизм преобразования виртуального адреса в физический при страничной организации памяти

Виртуальный адрес при страничном распределении может быть представлен в виде пары (*p*, *s*), где *p* – номер виртуальной страницы процесса (нумерация страниц начинается с 0), *s* – смещение в пределах виртуальной страницы. Учитывая, что размер страницы равен 2 в степени *k*, смещение *s* может быть получено простым отделением *k* младших разрядов в двоичной записи виртуального адреса. Оставшиеся старшие разряды представляют собой двоичную запись номера страницы *p*.

При каждом обращении к оперативной памяти аппаратными средствами выполняются следующие действия:

1. На основании начального адреса таблицы страниц (содержимое регистра адреса таблицы страниц), номера виртуальной страницы (старшие разряды виртуального адреса) и длины записи в таблице страниц (системная константа) определяется адрес нужной записи в таблице.

2. Из этой записи извлекается номер физической страницы.

3. К номеру физической страницы присоединяется смещение (младшие разряды виртуального адреса).

Использование в пункте (3) того факта, что размер страницы равен степени 2, позволяет применить операцию конкатенации (присоединения) вместо более длительной операции сложения, что уменьшает время

получения физического адреса, а значит, повышает производительность компьютера.

На производительность системы со страничной организацией памяти влияют временные затраты, связанные с обработкой страничных прерываний и преобразованием виртуального адреса в физический. При часто возникающих страничных прерываниях система может тратить большую часть времени впустую, на перемещение страниц. Чтобы уменьшить частоту страничных прерываний, следовало бы увеличивать размер страницы. Кроме того, увеличение размера страницы уменьшает размер таблицы страниц, а значит, уменьшает затраты памяти. С другой стороны, если страница велика, значит велика и фиктивная область в последней виртуальной странице каждой программы. В среднем на каждой программе теряется половина объема страницы, что в сумме при большой странице может составить существенную величину. Время преобразования виртуального адреса в физический в значительной степени определяется временем доступа к таблице страниц. В связи с этим таблицу страниц стремятся размещать в «быстрых» запоминающих устройствах. Это может быть, например, набор специальных регистров или память, использующая для уменьшения времени доступа ассоциативный поиск и кэширование данных.

Страничное распределение памяти может быть реализовано в упрощенном варианте, без выгрузки страниц на диск. В этом случае все виртуальные страницы всех процессов постоянно находятся в оперативной памяти. Такой вариант страничной организации хотя и не предоставляет пользователю виртуальной памяти, но почти исключает фрагментацию за счет того, что программа может загружаться в несмежные области, а также того, что при загрузке виртуальных страниц никогда не образуется остатков.

Сегментное распределение

При страничной организации виртуальное адресное пространство процесса делится механически на равные части. Это не позволяет дифференцировать способы доступа к разным частям программы (сегментам), а это свойство часто бывает очень полезным. Например, можно запретить обращаться с операциями записи и чтения в кодовый сегмент программы, а для сегмента данных разрешить только чтение. Кроме того, разбиение программы на «осмысленные» части делает принципиально возможным разделение одного сегмента несколькими процессами. Например, если два процесса используют одну и ту же математиче-

скую подпрограмму, то в оперативную память может быть загружена только одна копия этой подпрограммы.

Рассмотрим, каким образом сегментное распределение памяти реализует эти возможности. Виртуальное адресное пространство процесса делится на **сегменты, размер которых определяется программистом** с учетом смыслового значения содержащейся в них информации. Отдельный сегмент может представлять собой подпрограмму, массив данных и т.п. Иногда сегментация программы выполняется по умолчанию компилятором.

При загрузке процесса часть сегментов помещается в оперативную память (при этом для каждого из этих сегментов операционная система подыскивает подходящий участок свободной памяти), а часть сегментов размещается в дисковой памяти. Сегменты одной программы могут занимать в оперативной памяти несмежные участки. Во время загрузки система создает **таблицу сегментов процесса** (аналогичную таблице страниц), в которой для каждого сегмента указывается начальный физический адрес сегмента в оперативной памяти, размер сегмента, правила доступа, признак модификации, признак обращения к данному сегменту за последний интервал времени и некоторая другая информация. Если виртуальные адресные пространства нескольких процессов включают один и тот же сегмент, то в таблицах сегментов этих процессов делаются ссылки на один и тот же участок оперативной памяти, в который данный сегмент загружается в единственном экземпляре.

Система с сегментной организацией функционирует аналогично системе со страничной организацией: время от времени происходят прерывания, связанные с отсутствием нужных сегментов в памяти, при необходимости освобождения памяти некоторые сегменты выгружаются, при каждом обращении к оперативной памяти выполняется преобразование виртуального адреса в физический. Кроме того, при обращении к памяти проверяется, разрешен ли доступ требуемого типа к данному сегменту.

Виртуальный адрес при сегментной организации памяти может быть представлен парой (g, s), где g – номер сегмента, а s – смещение в сегменте. Физический адрес получается путем сложения начального физического адреса сегмента, найденного в таблице сегментов по номеру g, и смещения s.

Недостатком данного метода распределения памяти является фрагментация на уровне сегментов и более медленное по сравнению со страничной организацией преобразование адреса.

Странично-сегментное распределение

Как видно из названия, данный метод представляет собой **комбинацию страничного и сегментного распределения памяти** и, вследствие этого, сочетает в себе достоинства обоих подходов. Виртуальное пространство процесса делится на сегменты, а каждый сегмент, в свою очередь, делится на виртуальные страницы, которые нумеруются в пределах сегмента. Оперативная память делится на физические страницы. Загрузка процесса выполняется операционной системой постранично, при этом часть страниц размещается в оперативной памяти, а часть на диске. Для каждого сегмента создаётся своя таблица страниц, структура которой полностью совпадает со структурой таблицы страниц, используемой при страничном распределении.

Для каждого процесса создаётся таблица сегментов, в которой указываются адреса таблиц страниц для всех сегментов данного процесса. Начальный адрес таблицы сегментов загружается в специальный регистр процессора, когда активизируется соответствующий процесс.

Виртуальный адрес при странично-сегментном распределении состоит из трёх частей (g, p, s), где g – номер сегмента, p – номер виртуальной страницы процесса, s – смещение в пределах виртуальной страницы. Трансляция виртуального адреса в физический с использованием таблиц сегментов и страниц начинается (рис. 4.17) с того, что на основании начального адреса таблицы сегментов (содержимое регистра адреса таблицы сегментов), номера сегмента (старшие разряды виртуального адреса) определяется базовый адрес соответствующей таблицы страниц для данного сегмента. А дальше происходит всё то же самое, что при страничном распределении. По найденному базовому адресу таблицы страниц, номеру виртуальной страницы p из таблицы страниц извлекается старшая часть физического адреса страницы (n), к которой присоединяется смещение s (младшая часть).

Процесс преобразования адресов посредством таблиц является достаточно длительным и, естественно, приводит к снижению производительности системы. С целью ускорения этого процесса используется специальная, полностью ассоциативная кэш-память (рис. 4.17), которая называется **буфером преобразования адресов TLB (translation lookaside buffer)**.

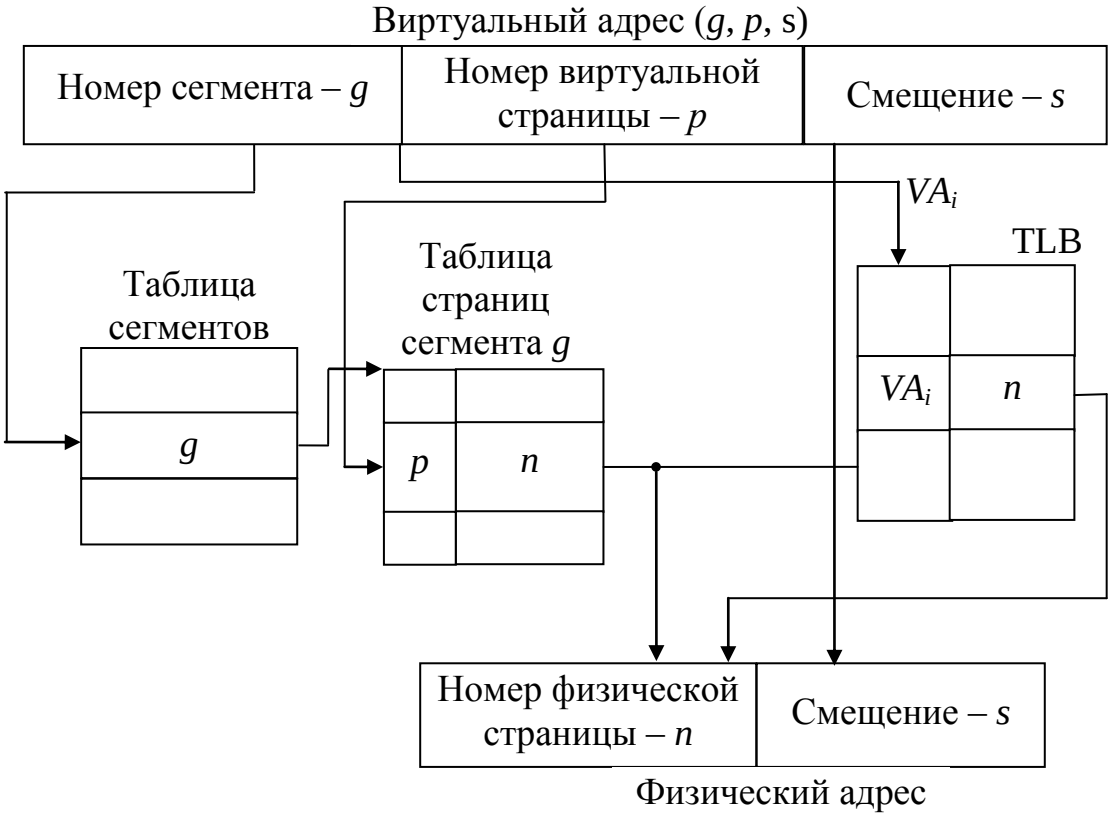


Рис. 4.17. Механизм преобразования адресов для странично-сегментной организации памяти с использованием TLB

Виртуальный адрес страницы VA_i, составленный из полей g и p, передаётся в TLB в качестве поискового признака (тега). Он сравнивается с тегами (*VA*) всех ячеек TLB, и при совпадении из найденной ячейки выбирается физический адрес страницы n, позволяющий сформировать полный физический адрес элемента данных, находящегося в ОП. Если совпадение не произошло, то трансляция адресов осуществляется обычными методами через таблицы сегментов и страниц. Эффективность преобразования адресов с использованием TLB зависит от коэффициента «попадания» в кэш-памяти, т.е. от того, насколько редко приходится обращаться к табличным методам трансляции адресов. Учитывая принцип локальности программ и данных, можно сказать, что при первом обращении к странице, расположенной в ОП, физический адрес определяется с помощью таблиц и загружается в соответствующую ячейку TLB. Последующие обращения к странице выполняются с использованием TLB.

4.4.5. Методы ускорения процессов обмена между ОП и ВЗУ

Эффективная скорость обмена между оперативным и внешним уровнями памяти в значительной степени определяется затратами на поиск секторов или блоков в накопителе ВЗУ. Для уменьшения влияния затрат времени поиска информации на скорость обмена используют традиционные методы **буферизации и распараллеливания.** Метод буферизации заключается в использовании так называемой дисковой кэш-памяти. **Дисковый кэш** уменьшает среднее время обращения к диску. Это достигается за счёт того, что копии данных, находящихся в дисковой памяти, заносятся в полупроводниковую память. Когда необходимые данные оказываются находящимися в кэше, время обращения значительно сокращается. За счет исключения задержек, связанных с позиционированием головок, время обращения может быть уменьшено в 2–10 раз.

Дисковый кэш может быть реализован программно или аппаратно.

Программный дисковый кэш – это буферная область в ОП, предназначенная для хранения считываемой с диска информации. При поступлении запроса на считывание информации с диска вначале производится поиск запрашиваемой информации в программном кэше.

При наличии в кэше требуемой информации она передаётся в процессор. Если она отсутствует, то осуществляется поиск информации на диске. Считанный с диска информационный блок заносится в буферную область ОП (программный дисковый кэш). Программа, управляющая дисковой кэш-памятью, осуществляет также слежение и за работой диска. Весьма хорошую производительность показывают программы Smart Drv, Ncache и Super PC-Kwik. Иногда для программного кэша используется дополнительная или расширенная память компьютера.

Аппаратный дисковый кэш – это встроенный в контроллер диска кэш-буфер с ассоциативным принципом адресации информационных блоков. По запросу на считывание информации вначале производится поиск запрашиваемого блока в кэше. Если блок находится в кэше, то он передаётся в ОП. В противном случае информационный блок считывается с диска и заносится в кэш для дальнейшего использования. При поступлении запроса на запись информационный блок из ОП заносится вначале в дисковый кэш и лишь затем (после выполнения соответствующих операций по поиску сектора) – на диск, при этом обычно копия блока в дисковом кэше сохраняется. Запись информационного блока из ОП в кэш производится на место блока, копия которого сохранена на диске. Для управления процессами копирования вводятся специальные указатели, которые определяют, сохранена ли данная копия на диске,

к какому информационному блоку обращение производилось ранее других и т.п. Копирование блока на диск производится по завершении операции поиска и не связано непосредственно с моментом поступления запроса.

Второй способ, позволяющий уменьшить снижение эффективной скорости обмена, вызванное операциями поиска на диске, связан с **использованием нескольких накопителей на диске.** Все информационные блоки распределяются по нескольким накопителям, причём так, чтобы суммарная интенсивность запросов по всем накопителям была одинаковой, а запросы по возможности чередовались. Если известны интенсивности запросов к каждому информационному блоку, то можно ранжировать эти блоки, а если при этом известны и логические связи между блоками, то связанные блоки с примерно одинаковыми интенсивностями запросов должны размещаться в разных накопителях. Это позволяет совместить операции обмена между ОП и одним из накопителей с операциями поиска очередного блока в других накопителях.

5. ОРГАНИЗАЦИЯ СИСТЕМНОГО ИНТЕРФЕЙСА И ВВОДА/ВЫВОДА ИНФОРМАЦИИ

5.1. Общая характеристика и классификация интерфейсов

Связь устройств ЭВМ друг с другом осуществляется с помощью интерфейсов.

Интерфейс представляет собой совокупность линий и шин, сигналов, электронных схем и алгоритмов (протоколов), предназначенных для осуществления обмена информацией между устройствами.

Производительность и эффективность использования компьютера определяются не только возможностями его процессора и пропускной способностью основной памяти, но в очень большой степени характеристиками интерфейсов, составом периферийных устройств (ПУ), их техническими данными.

Объединение отдельных подсистем (устройств, модулей) ЭВМ в единую систему основывается на многоуровневом принципе с унифицированным сопряжением между всеми уровнями **– стандартными интерфейсами**. Под стандартными интерфейсами понимают такие интерфейсы, которые приняты и рекомендованы в качестве обязательных отраслевыми или государственными стандартами, различными международными комиссиями, а также крупными зарубежными фирмами.

Интерфейсы характеризуются следующими параметрами:

- **пропускной способностью интерфейса** – количеством информации, которое может быть передано через интерфейс в единицу времени;
- **максимальной частотой передачи информационных сигналов** через интерфейс;
- **информационной шириной интерфейса** – числом бит или байт данных, передаваемых параллельно через интерфейс;
- **максимально допустимым расстоянием между соединяемыми устройствами;**
- **динамическими параметрами интерфейса** – временем передачи отдельного слова или блока данных с учётом продолжительности процедур подготовки и завершения передачи;
- **общим числом проводов** (линий) в интерфейсе.

Можно выделить следующие четыре классификационных признака интерфейсов:

- **способ соединения компонентов системы** (радиальный, магистральный, смешанный);

- **способ передачи информации** (параллельный, последовательный, параллельно-последовательный);
- **принцип обмена информацией** (асинхронный, синхронный);
- **режим передачи информации** (двусторонняя поочередная передача, односторонняя передача).

Радиальный интерфейс (рис. 5.1) даёт возможность всем модулям ($М_1$, …, $М_n$) работать независимо с центральным модулем (ЦМ). Он позволяет получить высокие скорости передачи информации, но требует большого количества шин. **Магистральный интерфейс** (общая шина) использует принцип разделения времени для связи между ЦМ и другими модулями. Он сравнительно прост в реализации, но лимитирует скорость обмена.

Параллельные интерфейсы позволяют передавать одновременно определенное количество бит или байт информации по многопроводной линии. **Последовательные интерфейсы** служат для последовательной передачи по двухпроводной линии.

В случае **синхронного интерфейса** моменты выдачи информации передающим устройством и приёма её в другом устройстве должны синхронизироваться, для этого используют специальную линию синхронизации. При **асинхронном интерфейсе** передача осуществляется по принципу «запрос-ответ». Каждый цикл передачи сопровождается последовательностью управляющих сигналов, которые вырабатываются передающим и приёмным устройствами. Передающее устройство может осуществлять передачу данных (байта или нескольких байтов) только после подтверждения приёмником своей готовности к приёму данных.

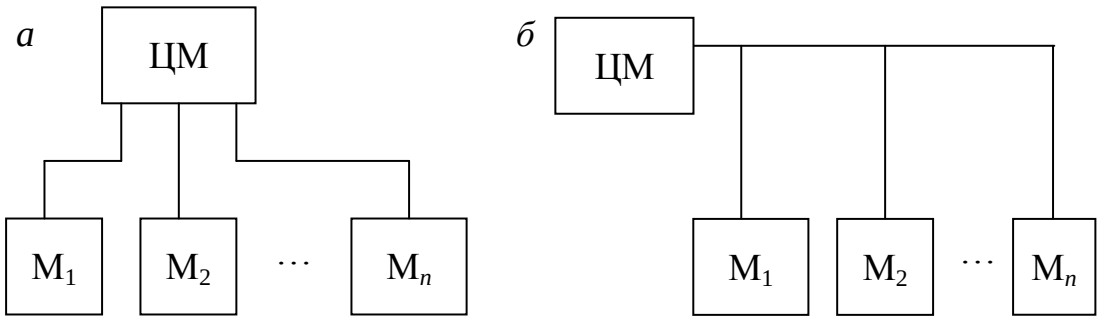


Рис. 5.1. Радиальный (*а*) и магистральный (*б*) интерфейсы

Классификация интерфейсов по назначению содержит следующие уровни сопряжений:

- **системные интерфейсы**;
- **локальные интерфейсы**;

- **интерфейсы периферийных устройств** (малые интерфейсы);
- **межмашинные интерфейсы**.

Системные интерфейсы предназначены для организации связей между центральным процессором, ОП и контроллерами (адаптерами) ПУ, а также между процессорами в многопроцессорных системах.

Локальные интерфейсы предназначены для организации связи с отдельными устройствами компьютера (видеокартой), а также для соединения микросхем чипсета между собой.

Назначение **интерфейсов периферийных устройств** (малых интерфейсов) состоит в выполнении функций сопряжения контроллера (адаптера) с конкретным механизмом ПУ.

Межмашинные интерфейсы используются в вычислительных системах и сетях.

Необходимость сохранения баланса производительности по мере роста быстродействия микропроцессоров привела к многоуровневой организации шин интерфейсов на основе использования **специализированных микросхем (чипсетов).**

Слово «чипсет» (chipset) в буквальном переводе означает «набор микросхем». Чипсет, который также называют **набором системной логики**, – это одна или две микросхемы, предназначенные для организации взаимодействия между процессором, памятью, интерфейсом графического адаптера, портами ввода/вывода и остальными компонентами компьютера. Со временем эти микросхемы стали называть **мостами,** появились устоявшиеся термины «северный мост» (North Bridge) и «южный мост» (South Bridge) чипсета. Если чипсет состоит из одной микросхемы, то такое решение называют одночиповым, а если из двух – двухмостовой схемой. В классической (традиционной) архитектуре двухмостового чипсета **северный мост** содержит контроллер памяти, контроллер графической шины (PCI Express), интерфейс взаимодействия с южным мостом и интерфейс взаимодействия с процессором через сокет определенного типа. Под **сокетом** понимается электрический соединитель, с помощью которого CPU компьютера соединяется с системной платой. Использование сокета позволяет при необходимости без особых проблем поменять процессор на более мощный из того же семейства. Сегодня для интеловских процессоров используются сокеты (разъемы) **в формате PGA** (pin grid array) для мобильных компьютеров и **LCA** (land grid array) – для настольных. В первом случае штыревые выводы, располагающиеся на нижней стороне корпуса процессора, устанавливаются в отверстия сокета. Во втором случае аналогично расположенные выводы процессора имеют вид плоских контактных пло-

щадок. При установке процессора в компьютер площадки CPU прижимаются к подпружиненным выводам сокета. Использование новой микроархитектуры процессоров, выпуск нового семейства CPU, повышение разрядности внешних шин и использование новых интерфейсов зачастую требуют смены сокета, а это, в свою очередь, влечёт за собой и замену чипсета.

На **южный мост** чипсета возлагается функция организации взаимодействия с устройствами ввода/вывода. Он содержит контроллеры жёстких дисков (SATA и/или PATA), сетевой контроллер, USB-контроллер, контроллер шин PCI и PCI Express, контроллер прерывания, DMA-контроллер, звуковой (аудио) контроллер. Кроме того, южный мост соединяется еще с одной важной микросхемой на материнской плате – микросхемой ROM-памяти BIOS (Basic Input-Output System – базовая система ввода/вывода). Это постоянная память, в которой хранится программа, отвечающая за базовые функции интерфейса и настройки оборудования, на котором она установлена. Наиболее широко среди пользователей компьютеров известна BIOS материнской платы, но BIOS присутствуют почти у всех компонентов компьютера: у видеоадаптеров, сетевых адаптеров, модемов, дисковых контроллеров, принтеров и т.д. Обозначение подобного базового программного обеспечения (ПО) термином «BIOS» присуще для компьютеров на базе процессоров с архитектурой x86. Для компьютеров на базе процессоров других типов для обозначения ПО, выполняющего подобные функции, используются другие термины, например, базовое ПО машин с процессором архитектуры SPARC называется PROM. Раньше к южному мосту подключалась еще одна микросхема Super I/O, которая отвечала за низкоскоростные порты RS232, LPT, RS/2. Сейчас эти функции выполняет южный мост. Для соединения северного и южного мостов друг с другом в большинстве случаев используются специальные локальные шины, причём разные производители применяют для этого разные шины с различной пропускной способностью (Intel – DMI, AMD – Alink Express, VIA – V-Link).

Чипсет является основой любой материнской платы. Фактически функциональность материнской платы и ее производительность на 90 % определяются именно чипсетом. От него зависят поддерживаемый тип процессора, тип памяти, тип сокета, а также функциональные возможности по подключению периферийных устройств. Основными компаниями на рынке чипсетов являются Intel, NVIDIA и AMD.

Шины процессора и памяти сравнительно короткие, обычно высокоскоростные и сбалансированные между собой для обеспечения мак-

симальной пропускной способности канала процессор–память. Шины ввода/вывода могут иметь большую протяжённость, поддерживать подсоединение многих типов устройств и обычно следуют одному из шинных стандартов. Обычно количество и типы устройств ввода/вывода в вычислительных системах не фиксируются (определяется количество разъёмов той или иной шины ввода/вывода), что даёт возможность пользователю самому подобрать необходимую конфигурацию. Шина ввода/вывода компьютера рассматривается как шина расширения, обеспечивающая постепенное наращивание устройств ввода/вывода. Поэтому стандарты играют огромную роль, позволяя разработчикам компьютеров и устройств ввода/вывода работать независимо.

5.2. Способы организации передачи данных

В подсистеме ввода/вывода ЭВМ используются три основных способа организации передачи данных между памятью и ПУ: программно-управляемая передача, передача по запросу прерывания от ПУ и прямой доступ к памяти (ПДП).

Программно-управляемая передача данных осуществляется при непосредственном участии и под управлением процессора, который при этом выполняет специальную подпрограмму ввода/вывода. Операция ввода/вывода инициируется центральным процессором, т.е. текущей командой программы. Данный способ является простым в реализации, но при обработке команды ввода/вывода ЦП бесполезно тратит время, ожидая готовности ПУ. Это значительно снижает производительность ЭВМ.

Второй способ передачи данных **по запросу прерывания от ПУ** реализуется под управлением контроллера прерываний (КПР) и позволяет организовывать более гибкое взаимодействие между ЦП и ПУ. Предположим, что в качестве ПУ используется клавиатура, предназначенная для ввода в ЭВМ команд, инструкций и данных. Каждый раз, когда пользователь (оператор) нажимает клавишу, ПУ выдает в КПР запрос на прерывание, который, в свою очередь, вырабатывает для ЦП сигнал прерывания. ЦП по этому сигналу приостанавливает работу текущей программы и передает управление подпрограмме ввода/вывода. Подпрограмма обрабатывает запрос и по её завершении ЦП возвращается к работе по текущей программе. Выполнение текущей программы продолжается до следующего нажатия клавиши, и далее процесс повторяется. В этом случае преимущество от использования прерывания очевидно (принципы работы системы прерывания программ описаны в разд. 2.6).

При программно-управляемой передаче данных ЦП на всё время этой передачи отвлекается от выполнения основной программы. Операция пересылки данных логически слишком проста, чтобы эффективно загружать логически сложную быстродействующую аппаратуру процессора. Вместе с тем при пересылке блока данных ЦП приходится для каждой единицы передаваемых данных (байт, слово) выполнять довольно много инструкций, чтобы обеспечить буферизацию данных, преобразование форматов, подсчёт количества переданных данных, формирование адресов в памяти и т.п. В результате скорость передачи данных при пересылке блока данных под управлением процессора оказывается недостаточной. Поэтому для быстрого ввода/вывода блоков данных и разгрузки ЦП от управления операциями ввода/вывода используют прямой доступ к памяти.

Прямой доступ к памяти

Прямой доступ к памяти (DMA – Direct Memory Access) – это такой способ обмена данными, который обеспечивает автономно от ЦП установление связи и передачу данных между ОП и ПУ. Прямой доступ к памяти освобождает процессор от управления операциями ввода/вывода, позволяет осуществлять параллельно во времени выполнение процессором программы с обменом данными между ОП и ПУ, производить этот обмен со скоростью, ограничиваемой только пропускной способностью ОП или ПУ.

Таким образом, ПДП, разгружая процессор от обслуживания ввода/вывода, способствует возрастанию общей производительности ЭВМ. Повышение предельной скорости ввода/вывода информации делает машину более приспособленной для работы в системах реального времени. Прямым доступом к памяти управляет **контроллер ПДП (DMA)** (рис. 5.2), который выполняет следующие функции:

1. Управление инициируемой процессором или ПУ передачей данных между ОП и ПУ.

2. Задание размера блока данных, который подлежит передаче, и области памяти, используемой при передаче.

3. Формирование адресов ячеек ОП, участвующих в передаче.

4. Подсчёт числа единиц данных (байт, слов), передаваемых от ПУ в ОП или обратно, и определение момента завершения заданной операции ввода/вывода.

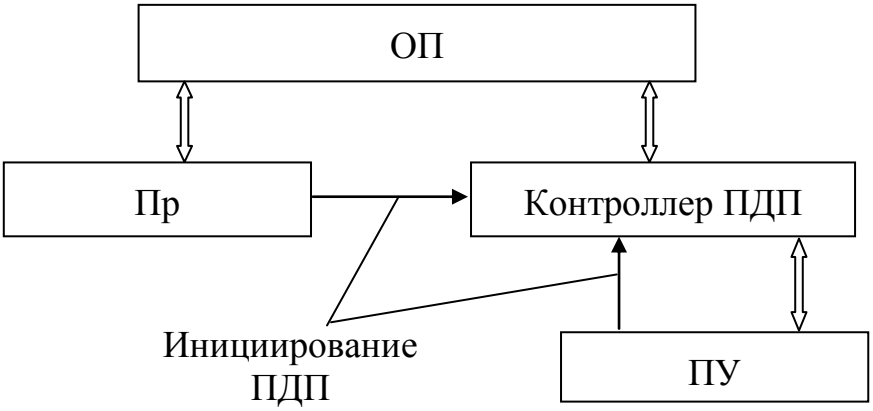


Рис. 5.2. Прямой доступ к памяти

ПДП обеспечивает высокую скорость обмена данными за счёт того, что управление обменом производится не программным путем, а аппаратурными средствами.

Контроллер ПДП обычно имеет более высокий приоритет в занятии цикла памяти по сравнению с процессором. Управление памятью переходит к контроллеру ПДП, как только завершится цикл ее работы, выполняемый для текущей команды процессора.

В современных ЭВМ используются все перечисленные способы передачи данных.

5.3. Системная организация настольных компьютеров на базе современных чипсетов компании Intel

Поскольку основным производителем процессоров для настольных компьютеров является корпорация Intel, то существуют семейства чипсетов под эти процессоры. В последнее время корпорации Intel удалось организовать практически полную монополию разработанных ею чипсетов для собственных процессоров. Бывшим лидерам рынка чипсетов, таким как VIA Technologies, SIS, NVIDIA, пришлось переориентироваться на разработку системной логики для других процессоров, например: AMD, VIA.

После перехода от микроархитектуры Net Burst к архитектуре Intel Core семейство чипсетов от Intel претерпело существенные изменения. Место на новых материнских платах заняла серия под кодовым именем Broadwater, которая в 2006 г. состояла из четырёх моделей: Intel Q965, Q963, G965 и P965. Эти чипсеты полностью поддерживали процессоры Core 2 Duo и работали на частоте системной шины FSB 1066 МГц.

Появившееся позже семейство чипсетов Bearlake (Intel X38, P35, G35, G33, Q35, Q33) пришло на смену предыдущего поколения микросхем и предназначалось для высокопроизводительных систем с процессорами, произведёнными по 45 нм техпроцессу.

Семейство чипсетов (Intel X58, P55, H55, H57) предназначалось для системной организации компьютеров на базе процессоров с микроархитектурой Nehalem.

С тех пор, как контроллер памяти и контроллер графической шины PCI Express переместились внутрь процессора, дизайн наборов вистемной логики существенно упростился. Чипсеты, состоящие ранее из пары микросхем – северного и южного мостов, переродились (начиная с Intel P55) в единый чип – концентратор, отвечающий за реализацию интерфейсов ввода-ввывода. И теперь их обновление не оказывает существенного влияния на производительность и возможности платформы, а сказывается лишь на конструкции материнских плат, комплектуемых тем или иным набором дополнительных контроллеров. Поэтому ожидать, что выход очередного поколения наборов логики может как-то существенно повлиять на потребительские характеристики систем, явно не следует.

Чипсеты шестой (P67, H67, Q67, Z68) и седьмой (Z77, Z75, H77) серий мало отличались друг от друга и использовались для процессоров Sandy Bridge и Ivy Bridge. Они поддерживали разъем LGA 1155.

5.3.1. Системная организация компьютеров на базе чипсетов Intel 8-й и 9-й серий

Чипсеты 8-й и 9-й серий по разводке, питанию, функциональным возможностям отличаются незначительно. Они используют для настольных систем один и тот же сокет LGA 1150.

Семейство чипсетов 8-й серии (Q87, Q85, B85, Z85, H87, H81) поддерживают процессоры Intel Core четвертого поколения с микроархитектурой Haswell и Haswell Refresh. К Haswell Refresh относятся процессоры Haswell, у которых увеличены на 100 МГц базовая и турбо частоты.

Чипсеты 9-й серии (Z97, H97) кроме процессоров четвертого поколения, упомянутых выше, поддерживают процессоры Intel Core пятого поколения (процессоры Broadwell с техпроцессом 14 нм).

Перечисленные чипсеты можно распределить по трем категориям. Для корпоративного сегмента используются чипсеты Q87, Q85. Для сегмента малого и среднего бизнеса – B85. На потребительском уровне

производительные (флагманские) чипсеты: Z87, Z97 PCH. Варианты чипсетов для массового рынка: H87, H97. Для компьютеров начального уровня используется чипсет H81.

В качестве примера более подробно рассмотрим (рис.5.3) системную организацию процессоров Intel 4-го и 5-го поколений на базе чипсета Z97.

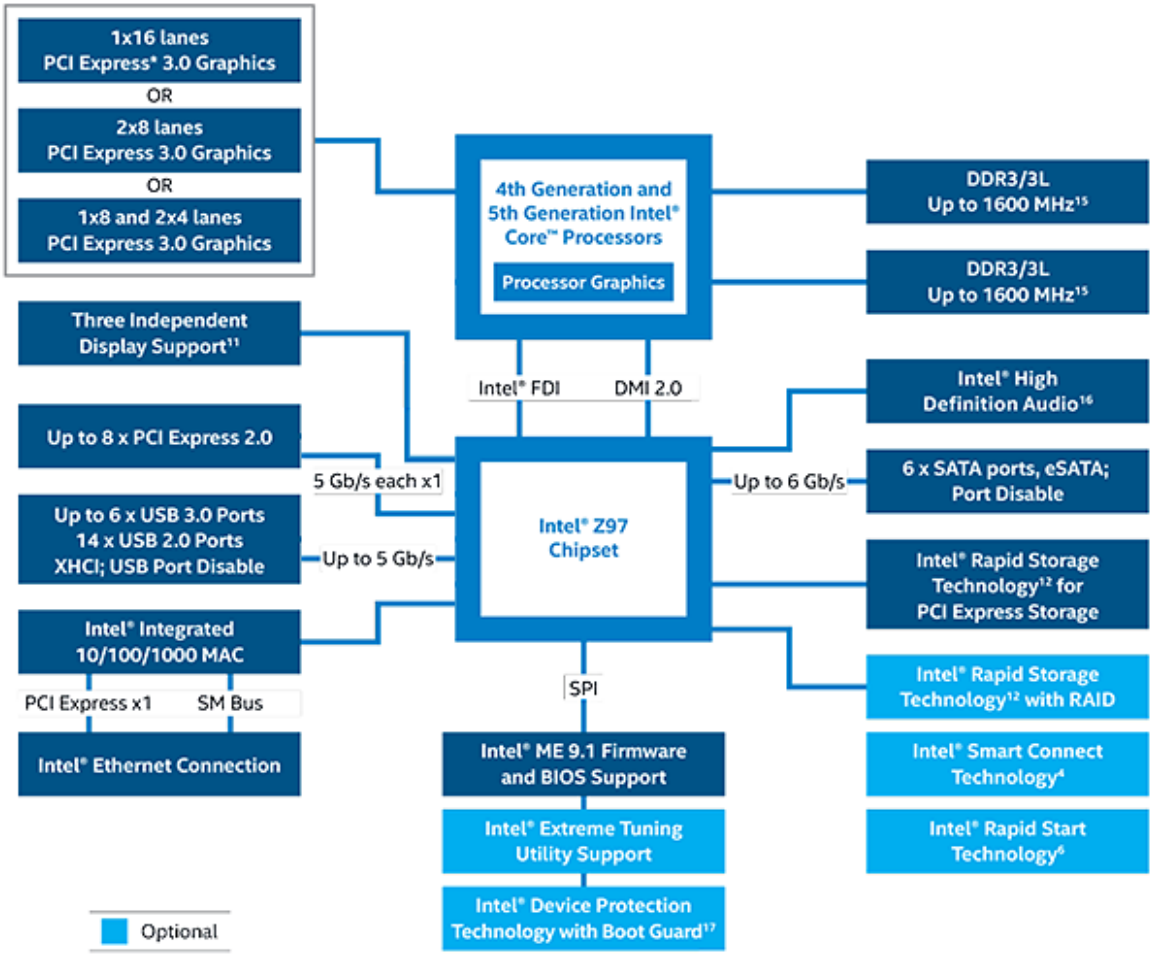


Рис. 5.3 Системная организация компьютера на базе чипсета Intel Z97

Поддержка функций процессора

Процессоры имеют несколько портов PEG (суммарно 16 линий PCI Express 3.0 Graphics), и в зависимости от примененного на плате чипсета эти порты могут комбинироваться по-разному для реализации различных вариантов слотов PCIe. Например, чипсет Intel Z97 (как и его аналог Intel Z87) позволяет использовать порты PEG в следующих комбинациях: x16, x8/x8 или x8/x4/x4. Таким образом, на платах с чипсетом Intel Z97 может быть реализован один слот PCIe x16, два слота PCIe x8 или один слот PCIe x8 и два слота PCIe x4. Чипсет Intel H97 допускает

только одну возможную комбинацию: x16 (то есть на платах с чипсетом Intel H97 может быть реализован только один слот PCIe x16).

Шины DMI и FDI

Для связи процессора с чипсетом Intel 9-й серии используется все та же полнодуплексная шина DMI 2.0 (Direct Media Interface) с пропускной способностью 20 Гбит/с в каждом направлении.

Чипсеты 9-й серии поддерживают возможность одновременного подключения до трех мониторов к процессорному графическому ядру, как и в случае чипсетов 8-й серии. Для этого используется шина Intel FDI (Flexible Display Interface), по которой процессорное графическое ядро соединяется с блоком дисплейного вывода в чипсете. Как и прежде, шина FDI использует две линии с пропускной способностью 2,7 Гбит/с каждая. Для реализации аналогового интерфейса VGA в чипсете есть интегрированный RAMDAC (180 МГц) – часть графического контроллера ответственного за преобразование пиксельных значений из цифровой в аналоговую форму.

Порты PCI Express, SATA и USB

Чипсеты Intel 9-й серии поддерживают до 8 портов PCI Express 2.0 (с пропускной способностью 5,0 Гбит/с). Кроме того, имеется в новых чипсетах и интегрированный SATA-контроллер, который обеспечивает поддержку до шести портов SATA (6 Гбит/с), причем каждый из шести SATA-портов может быть реализован как порт eSATA. Естественно, поддерживается технология Intel RST (Rapid Storage Technology), которая позволяет конфигурировать SATA-контроллер в режиме RAID-контроллера с поддержкой уровней 0, 1, 5 и 10.

Как и прежде, в чипсеты Intel 9-й серии интегрирован один USB-контроллер xHCI (eXtensible Host Controller Interface), который поддерживает 14 логических портов USB 2.0 и 6 логических портов USB 3.0. В пересчете на физические порты, получаем 14 разъемов, до шести из которых поддерживают USB 3.0, а остальные — только USB 2.0. Кроме того, имеется в чипсете и два контроллера EHCI (Enhanced Host Controller Interface), один из которых поддерживает 8 портов USB 2.0, а второй — 6 портов USB 2.0 (в сумме опять-таки получаем, что имеется 14 логических портов USB 2.0). Понятно, что имеются и программно-управляемые коммутаторы, которые позволяют переключать физические порты USB между контроллерами EHCI и xHCI. Говоря проще (без привязки к контроллерам), чипсеты Intel 9-й серии, как и прежде, могут

организовать 14 USB-портов, из которых до 6 портов могут быть USB 3.0, а остальные — USB 2.0.

Технология Flexible IO

Опять же, как и в чипсетах 8-й серии, в чипсетах 9-й серии реализована поддержка технологии Flexible IO, которая позволяет конфигурировать высокоскоростные порты ввода/вывода (PCIe, SATA, USB 3.0), убирая одни порты и добавляя другие.

Работает это следующим образом. Всего в чипсете имеется 18 высокоскоростных портов ввода/вывода. Причем конфигурация 14 портов строго фиксирована: 4 порта USB 3.0, 6 портов PCIe 2.0 и 4 порта SATA (6 Гбит/с). А еще 4 порта можно переконфигурировать: 2 из них могут работать либо как USB 3.0, либо как PCIe, а 2 других — как PCIe или SATA. При этом общее количество портов PCIe не может превышать восьми.

Собственно, именно поэтому мы говорим, что чипсет поддерживает до 8 портов PCIe, до 6 портов USB 3.0 и до 6 портов SATA (6 Гбит/с), но это не значит, что их можно реализовать одновременно. Всего высокоскоростных портов ввода/вывода не может быть более 18. Поэтому если на плате реализовано 6 портов USB 3.0 и 6 портов SATA, то на ней может быть только 6 портов PCIe. Если же реализовано 6 портов SATA и 8 портов PCIe, то может быть только 4 порта USB 3.0.

Если говорить о других функциях чипсетов 9-й серии, интересных с точки зрения пользователя, то стоит отметить наличие интегрированного сетевого гигабитного контроллера (MAC-уровня). Естественно, имеется интегрированный аудиоконтроллер Intel HD Audio (как и ранее). Также поддерживаются все те же технологии, которые поддерживались чипсетами 8-й серии: Intel Trusted Execution Technology (Intel TXT), Intel Anti-Theft Technology (Intel AT), Intel Virtualization Technology for Directed I/O (Intel VT-d) и пр.

И единственная новая функция, которой не было в чипсетах 8-й серии — это поддержка накопителей PCIe SSD (PCIE NAND Technology for PCIE SSD support). Собственно, речь идет о поддержке чипсетом интерфейса PCIe M.2. Причем в данном случае нет никакого отдельного контроллера — данная поддержка реализована на уровне конфигурирования регистров чипсета. Отметим, что накопителей с интерфейсом PCIe M.2. пока еще на рынке очень мало, да и стоят они дорого.

5.3.2. Системная организация компьютеров на базе чипсетов Intel 100-й серии

Семейство чипсетов 100-й серии для настольных вариантов Skylake (шестое поколение Intel Core) включает в себя как минимум 6 модификаций: Q170, Q150, B150, Z170, H170, H110, различающихся по позиционированию и характеристикам. Чипсеты распределены по трем категориям. Для корпоративного сегмента используются чипсеты Q150 и Q170, которые заменили Q85 и Q87. Для сегмента малого и среднего бизнесса – чипсет B150 PCH, заменяющий южный мост B85. На потребительском уровне новый чипсет Z170 идет на смену предыдущего флагмана Z97 PCH, дополняя Q170. Вариант для массового рынка H170 заменит H97, а на начальном уровне чипсет H81 уступит место H110.

Одновременно с появлением 100-й серии чипсетов в обиход входит новый сокет – LGA 1151. По габбаритам, внешнему виду и числу контактов он почти не отличатеся от предыдущей верии LGA 1150. Электрически, LGA 1151 имеет значительные отличия. Они связаны с появлением в Skylake поддержки памяти стандарта DDR4, удалением из процессора встроенного преобразователя питания (FIVR) и внедрением новой скоростной шины, связывающей CPU с набором системной логики.

Прошлые интеловские чипсеты, применяющиеся с процессорами поколений Haswell и Broadwell, нередко вызывают нарекания из-за того, что они и сами не предоставляют достаточное число высокоскоростных интерфейсов, и серьёзно ограничивают возможности по их добавлению в систему через внешние контроллеры. Корни этой проблемы — в шине DMI 2.0, связывающей CPU с набором логики. Её пропускная способность составляет всего 2 Гбайт/с (в каждую сторону), что ограничивает полосу пропускания, которую могут получать в своё распоряжение подключаемые через чипсет устройства.

В процессорах Skylake для настольных систем Intel наконец реализовала новую версию этой шины – DMI 3.0, которая теперь базируется на протоколе PCI Express 3.0 и имеет увеличенную до 3,9 Гбайт/с (в каждую сторону) пропускную способность. Такое изменение стало хорошим фундаментом для пересмотра функциональности наборов логики, и благодаря этому чипсеты для Skylake, имеющие кодовое имя Sunrise Point и относящиеся к сотой серии, получили заметно более широкий набор возможностей.

В качестве примера рассмотрим системную организацию процессоров на базе чипсета Intel Z170. Как видно из рисунка 5.4 этот набор

логики принциально отличается от своих предшественников появлением поддержки шины PCI Express 3.0 (8 Гбит/с) и увеличением количества портов USB 3.0. Однако, к сожалению, пока Intel воздержалась от интеграции в набор логики контроллера USB 3.1 и такие высокоскоростные порты, очевидно, будут реализовываться на платах через дополнительные чипы. Впрочем, в этом нет особой проблемы – число поддерживаемых линий PCI Express 3.0 в Intel Z170 выросло до 20, и этого должно с лихвой хватить и на всякие добавочные контроллеры, и на разветвлённые скоростные интерфейсы для подключения твердотельных накопителей.

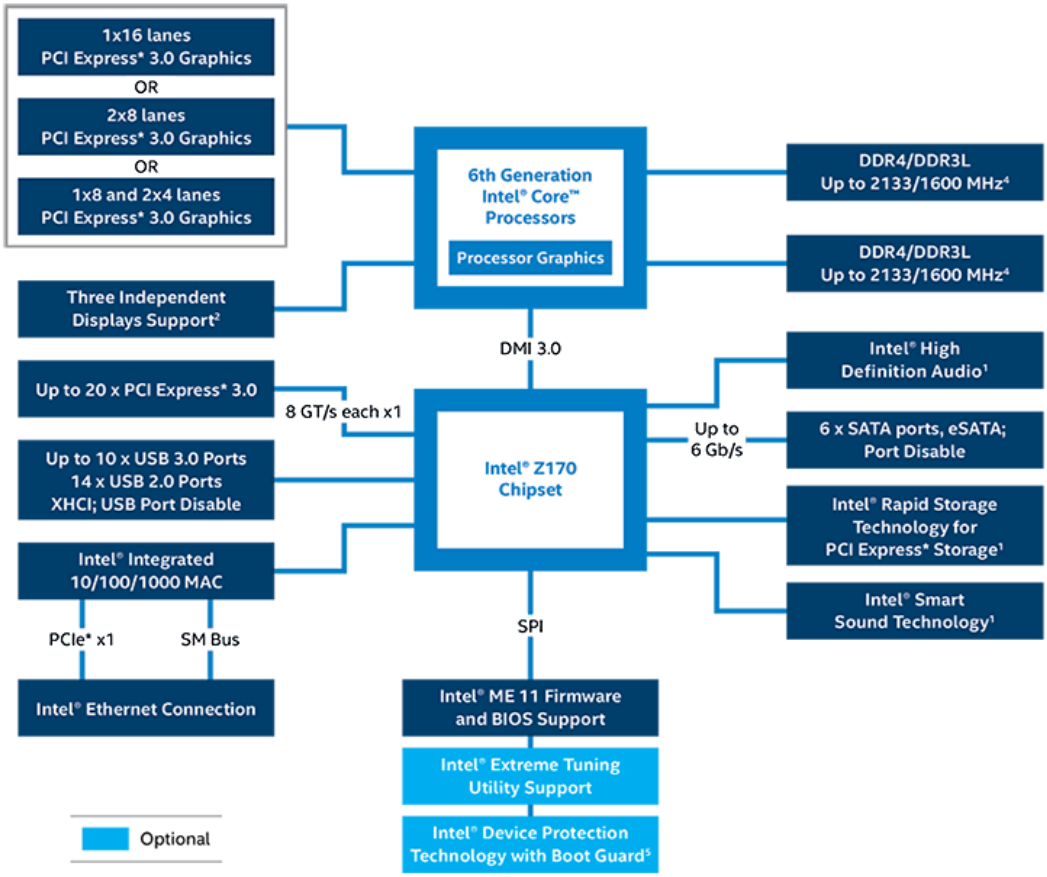


Рис. 5.4 Системная организация компьютера на базе чипсета Intel Z170

Исходя из потребностей современных интерфейсов, у типичной материнской платы на базе Z170 отведено по четыре линии PCI Express 3.0 на каждый разъём M.2 или порт U.2, по две линии – на каждую пару USB 3.1-портов и по одной линии – для каждого гигабитного LAN-контроллера. Но даже исходя из этой арифметики понятно, что 20 линий PCI Express 3.0 хватит не только на оснащение материнской платы всеми необходимыми дополнительными интерфейсами, но и на доба-

вочные слоты PCIe 3.0 x4 или даже PCIe 3.0 x8. Иными словами, благодаря набору логики Intel Z170, производители LGA1151-плат могут придать им заметно более широкие возможности по сравнению с материнскими платами, предназначенными для Haswell или Broadwell, не прибегая к ухищрениям вроде концентраторов PCI Express.

Кстати, не стоит забывать, что помимо 20 линий PCI Express 3.0, набор логики Z170 предлагает также шесть традиционных портов SATA (6 Гбит/с) и 10 портов USB 3.0, для реализации которых никакие дополнительные чипы вообще не требуются.

Характеристики чипсетов 100-й серии приведены в таблице 5.1.

Таблица 5.1

Характеристики чипсетов 100-й серии

	Feature/ Capability	Q170	Q150	B150	H110	H170	Z170
CHIPSET I/O	Chipset PCI Express* Gen 3 Lanes	Up to 20	10	8	6 (Gen 2 Only)	Up to 16	Up to 20
	SATA Gen 3	Up to 6	Up to 6	Up to 6	4	Up to 6	Up to 6
	USB 3.0	Up to 10	Up to 8	6	4	Up to 8	Up to 10
	Total USB Ports (USB 2.0 + 3.0)	14	14	12	10	14	14
	SATA Express Capable Ports (x2)	Up to 3	Up to 1	Up to 1	0	Up to 2	Up to 3
	Intel® RST for PCIe Storage Ports (x4 M.2 or x2 SATA Express)	Up to 3	0	0	0	Up to 2	Up to 3
	Enhanced SPI	✓	✓	✓		✓	✓
CPU	Processor PCI Express* Gen 3 1x16 Port	X4, x8, x16	1x16	1x16	1x16	1x16	X4, x8, x16

На этом преимущества новых наборов логики не заканчиваются. Важное обновление затронуло и технологию Intel Rapid Storage, возможности которой в LGA1151-чипсетах существенно расширились. Так, в ней появилась поддержка NVMe-накопителей, а, кроме того, объединять в RAID-массивы теперь можно и SSD, подключенные в систему по шине PCI Express через разъёмы SATA Express, M.2, U.2 или напрямую.

Таким образом, процессоры Skylake привлекательны не только новой микроархитектурой с возросшей удельной производительностью и поддержкой более скоростной и прогрессивной памяти, но и тем, что вся платформа в целом стала заметно лучше и функциональнее.

6. МНОГОПРОЦЕССОРНЫЕ И МНОГОМАШИННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

6.1. Архитектуры вычислительных систем

Точно также, как однопроцессорные компьютеры представлены по классификации М. Флина архитектурами с одним потоком данных SISD и множеством потоков данных SIMD, так и многопроцессорные системы могут быть представлены двумя базовыми типами архитектур в зависимости от параллелизма данных:

- **MISD (Multiple Instruction Single Data)** – множество потоков команд – один поток данных;
- **MIMD (Multiple Instruction Multiple Data) –** множество потоков команд – множество потоков данных.

Класс MISD долгое время пустовал, поскольку не существовало практических примеров реализации систем, в которых одни и те же данные обрабатываются большим числом параллельных процессов. В дальнейшем для MISD нашлась адекватная организация вычислительной системы – распределённая мультипроцессорная система с общими данными. Наиболее простая и самая распространённая система этого класса – обычная локальная сеть персональных компьютеров, работающая с единой базой данных, когда много процессоров обрабатывают один поток данных. Впрочем, тут есть одна тонкость. Как только в такой сети все пользователи переключаются на обработку собственных данных, недоступных для других абонентов сети, MISD-система превращается в систему с множеством потоков команд и множеством потоков данных, соответствующую MIMD-архитектуре.

Так как только MIMD-архитектура включает все уровни параллелизма – от конвейера операций до независимых заданий и программ, то любая вычислительная система этого класса в частных приложениях может выступать как SISD- и SIMD-система. Например, если многопроцессорный комплекс выполняет одну единственную программу без каких-либо признаков векторного параллелизма данных, то в этом конкретном случае он функционирует как обычный SISD-компьютер, и весь его потенциал остается невостребованным. Таким образом, употребляя термин «MIMD», надо иметь в виду не только много процессоров, но и множество вычислительных процессов, одновременно выполняемых в системе.

Другая классификация многопроцессорных вычислительных систем (МВС) основана на разделении МВС по двум критериям: **способу построения памяти** (общая или распределенная) и **способу передачи информации**. Основные типы машин представлены в табл. 6.1. Здесь приняты следующие обозначения: *P* – элементарный процессор, *M* – элемент памяти, *К* – коммутатор, *С* – кэш-память.

Параллельная вычислительная система с общей памятью и шинной организацией обмена (машина 1) позволяет каждому процессору системы «видеть», как решается задача в целом, а не только те части, над которыми он работает. Общая шина, связанная с памятью, вызывает серьёзные проблемы для обеспечения высокой пропускной способности каналов обмена. Одним из способов обойти эту ситуацию является использование кэш-памяти (машина 2). В этом случае возникает проблема когерентности (адекватности) содержимого кэш-памяти и основной памяти. Другим способом повышения производительности систем является отказ от общей памяти (машина 3).

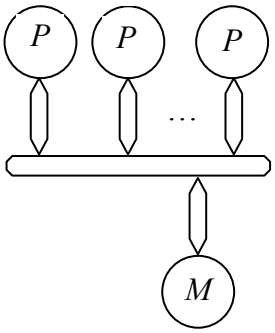
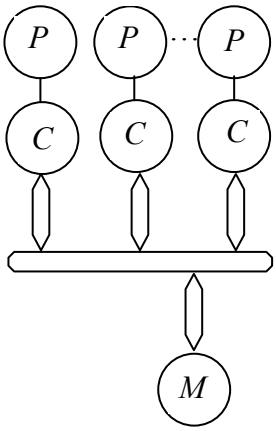
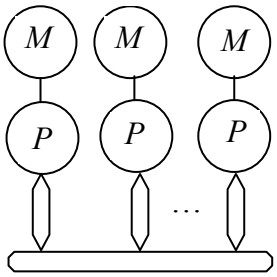
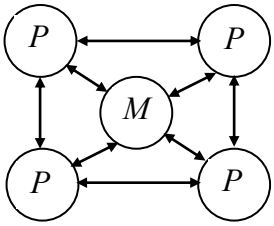
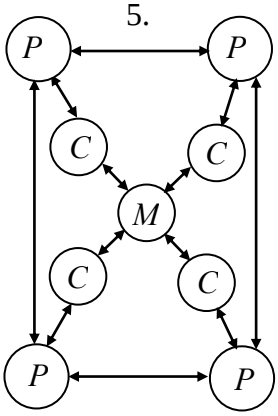
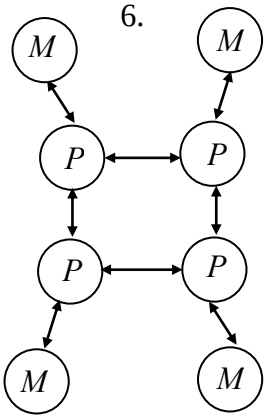
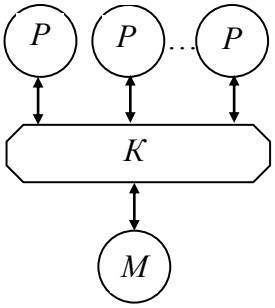
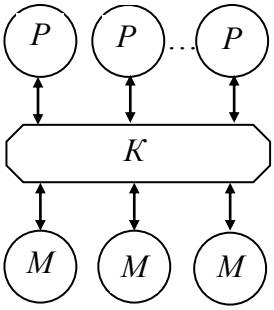
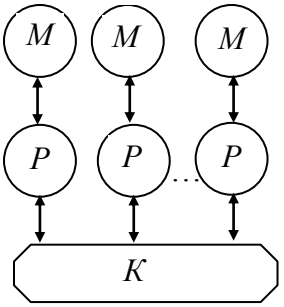
Идеальной машиной является вычислительная система, у которой каждый процессор имеет прямые каналы связи с другими процессорами, но в этом случае требуется чрезвычайно большой объём оборудования для организации межпроцессорных обменов. Определенный компромисс представляет сеть с фиксированной топологией, в которой каждый процессор соединен с некоторым подмножеством процессоров системы (машины 4, 5, 6).

Если процессорам, не имеющим непосредственного канала обмена, необходимо взаимодействовать, они передают сообщения через промежуточные процессоры. Одно из преимуществ такого подхода – не ограничивается рост числа процессоров в системе. Недостаток – требуется оптимизация прикладных программ, чтобы обеспечить выполнение параллельных процессов, для которых необходимо активное воздействие на соседние процессоры.

Наиболее интересным вариантом для перспективных параллельных вычислительных комплексов является сочетание достоинства архитектур с распределенной памятью и каналами межпроцессорного обмена. Один из возможных методов построения таких комбинированных архитектур – конфигурация с коммутацией, когда процессор имеет локальную память, а соединяются процессоры между собой с помощью коммутатора (машина 9). Коммутатор может оказаться весьма полезным для группы процессоров с распределяемой памятью (машина 8). Данная конфигурация похожа на машину с общей памятью (машина 7), но здесь исключены проблемы пропускной способности шины.

Таблица 6.1

Основные типы машин

Типы передачи сообщений	Типы памяти		
	Общая память	Общая и распределенная	Распределенная память
Шинные соединения	1. 	2. 	3. 
Фиксированные перекрестные соединения	4. 	5. 	6. 
Коммутационные структуры	7. 	8. 	9. 

MIMD-системы по способу взаимодействия процессоров (рис. 6.1) делятся на **системы с сильной и слабой связью.**

Системы с сильной связью (иногда их называют «истинными» мультипроцессорами) основаны на объединении процессоров на общем поле оперативной памяти.

Системы со слабой связью представляются многопроцессорными и многомашинными системами с распределенной памятью. Разница организации MIMD-систем с сильной и слабой связью проявляется при обработке приложений, отличающихся интенсивностью обменов между процессами.

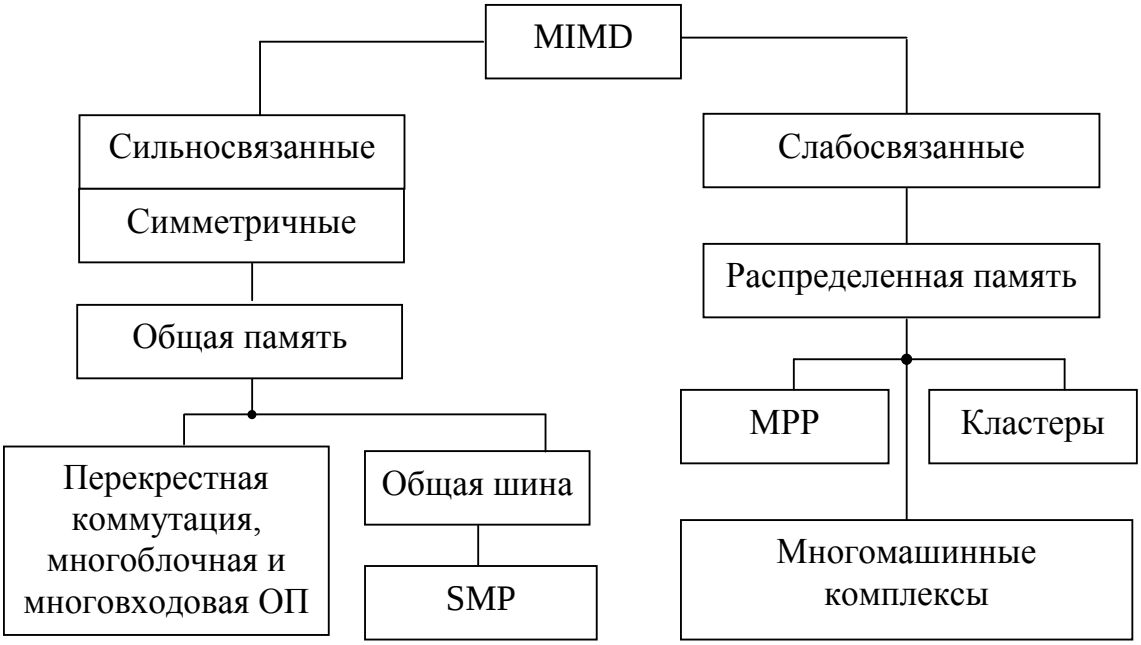


Рис. 6.1. Классификация вычислительных систем с MIMD-архитектурой

6.2. Сильносвязанные многопроцессорные системы

В архитектурах многопроцессорных сильносвязанных систем можно отметить две важнейшие характеристики: **симметричность** (равноправность) всех процессоров системы и **распределение** всеми процессорами **общего поля оперативной памяти.**

В таких системах, как правило, число процессоров невелико (не больше 16) и управляет ими централизованная операционная система. Процессоры обмениваются информацией через общую оперативную память. При этом возникают задержки из-за межпроцессорных конфликтов. При создании больших мультипроцессорных ЭВМ (мэйнфреймов, суперЭВМ) предпринимаются огромные усилия по увеличе-

нию пропускной способности оперативной памяти (перекрестная коммутация, многоблочная и многовходовая оперативная память и т.д.). В результате аппаратные затраты возрастают чуть ли не в квадратичной зависимости, а производительность системы упорно «не желает» увеличиваться пропорционально числу процессоров. То, что могут себе позволить дорогостоящие и сложные мэйнфреймы и суперкомпьютеры, не годится для компактных многопроцессорных серверов.

Архитектура SMP

Для простой и «дешевой» поддержки многопроцессорной организации была предложена **архитектура SMP –** мультипроцессирование с разделением памяти, предполагающая объединение процессоров на общей шине оперативной памяти. За аппаратную простоту реализации средств SMP приходится расплачиваться процессорным временем ожидания в очереди к шине оперативной памяти. В большинстве случаев пользователи готовы добавить в сервер один или более процессоров (но редко – более четырёх) в надежде увеличить производительность системы. Стоимость этой операции ничтожна по сравнению со стоимостью всего сервера, а результат чаще всего оправдывает ожидания пользователя.

Пропускную способность памяти в таких системах можно значительно увеличить путём применения больших многоуровневых кэшей. При этом кэши могут содержать как **разделяемые**, так и **частные данные**. Частные данные – это данные, которые используются одним процессором, в то время как разделяемые данные используются многими процессорами, по существу обеспечивая обмен между ними. Если кэшируются разделяемые данные, то они реплицируются и могут содержаться в нескольких кэшах. Кроме сокращения задержки доступа и требуемой полосы пропускания, такая репликация данных способствует также общему сокращению количества обменов. Однако кэширование разделяемых данных вызывает новую проблему: **когерентность кэш-памяти.** Эта проблема возникает из-за того, что значение элемента данных в памяти, используемое двумя разными процессорами, доступно этим процессорам только через их индивидуальные кэши. На рис. 6.2 показан простой пример, иллюстрирующий эту проблему, где A' и B' – кэшированные копии элементов A и B в основной памяти. Когерентное (адекватное) состояние кэша и основной памяти, когда $A' = A$ & $B' = B$, изображено на рис. 6.2, *а*. Во втором случае (рис. 6.2, *б*) предполагается использование кэш-памяти с обратной записью, когда ЦП записывает

значение 550 в ячейку A'. В результате A' содержит новое значение, а в основной памяти осталось старое значение – 100. При попытке вывода A из памяти будет получено старое значение.

В третьем случае (рис. 6.2, *в*) подсистема ввода/вывода вводит в ячейку памяти B новое значение 440, а в кэш-памяти осталось старое значение B.

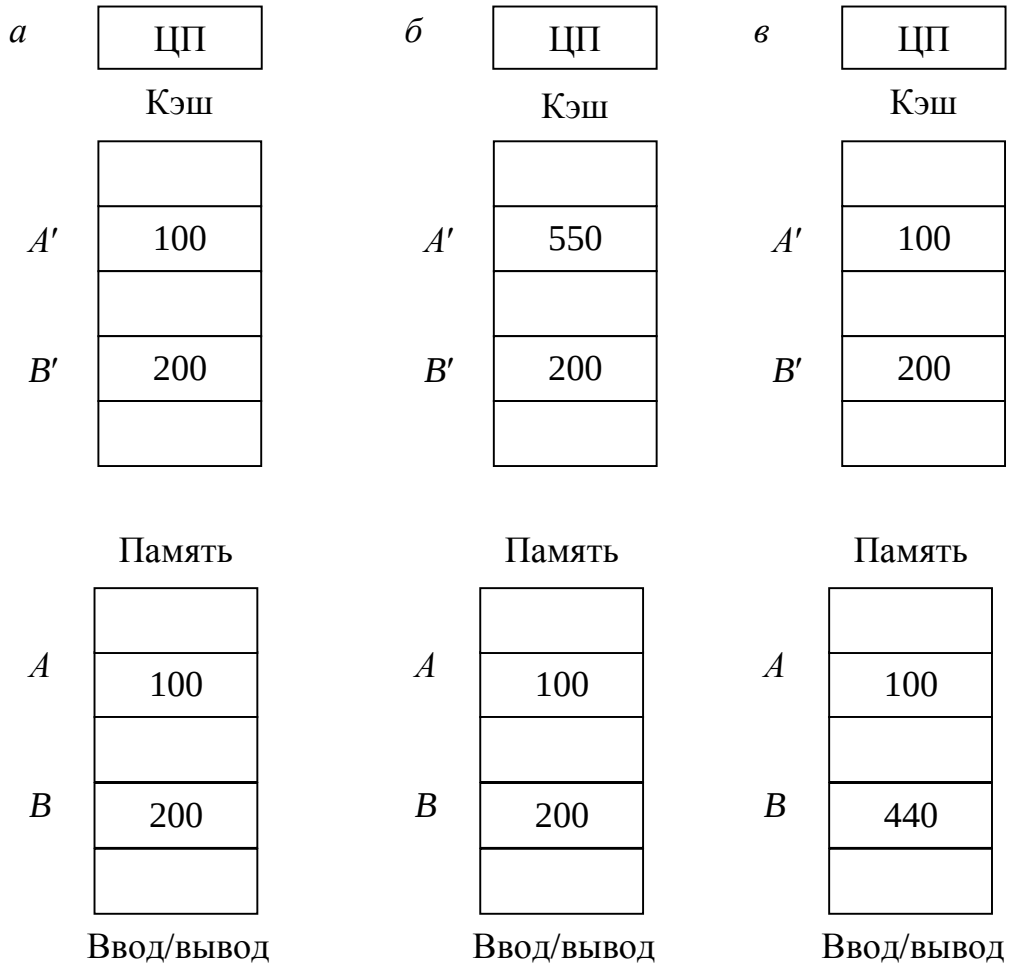


Рис. 6.2. Кэш и память когерентны: $A' = A$ & $B' = B$ (*а*), кэш и память некогерентны: $A' \neq A$ (*б*), кэш и память некогерентны: $B' \neq B$ (*в*)

Обычно в малых мультипроцессорах (с небольшим количеством процессоров) используется аппаратный механизм, называемый **протоколом когерентности кэш-памяти**, позволяющий решить эту проблему.

Основное преимущество SMP – **относительная простота программирования**. В ситуации, когда все процессоры имеют одинаково быстрый доступ к общей памяти, вопрос о том, какой из процессоров

какие вычисления будет выполнять, не столь принципиален, и значительная часть вычислительных алгоритмов, разработанных для последовательных компьютеров, может быть ускорена с помощью распараллеливающих и векторизирующих трансляторов.

Архитектура SMP стала своего рода стандартом для всех современных многопроцессорных серверов.

6.3. Слабосвязанные многопроцессорные системы

Существует несколько способов построения крупномасштабных систем с распределённой памятью.

1. Многомашинные системы. В таких системах отдельные компьютеры объединяются либо с помощью сетевых средств, либо с помощью общей внешней памяти (обычно – дисковые накопители большой емкости).

2. Системы с массовым параллелизмом MPP (Massively Parallel Processor). Идея построения систем этого класса тривиальна: берутся серийные микропроцессоры, снабжаются каждый своей локальной памятью, соединяются посредством некоторой коммуникационной среды, например сетью.

Системы с массовым параллелизмом могут содержать десятки, сотни и тысячи процессоров, объединённых коммутационными сетями самой различной формы – от простейшей двумерной решетки до гиперкуба. Достоинства такой архитектуры: во-первых, она использует стандартные микропроцессоры; во-вторых, если требуется высокая терафлопсная производительность, то можно добавить в систему необходимое количество процессоров; в-третьих, если ограничены финансы или заранее известна требуемая вычислительная мощность, то легко подобрать оптимальную конфигурацию.

Однако есть и решающий «минус», сводящий многие «плюсы» на нет. Дело в том, что межпроцессорное взаимодействие в компьютерах этого класса идет намного медленнее, чем происходит локальная обработка данных самими процессорами. Именно поэтому написать эффективную программу для таких компьютеров очень сложно, а для некоторых алгоритмов иногда просто невозможно.

3. Кластерные системы. Данное направление, строго говоря, не является самостоятельным, а, скорее, представляет собой комбинацию из архитектур SMP и MPP. Из нескольких стандартных микропроцессоров и общей для них памяти формируется вычислительный узел (обыч-

но по архитектуре SMP). Для достижения требуемой вычислительной мощности узлы объединяются высокоскоростными каналами.

Эффективность распараллеливания процессов во многих случаях сильно зависит от топологии соединения процессорных узлов. Идеальной является топология, в которой любой узел мог бы напрямую связаться с любым другим узлом. Однако в кластерных системах это технически трудно реализуемо. Обычно процессорные узлы в современных кластерных системах образуют или двумерную решетку, или гиперкуб.

Для синхронизации параллельно выполняющихся в узлах процессов необходим обмен сообщениями, которые должны доходить из любого узла системы в любой другой узел. При этом важной характеристикой является максимальное расстояние между узлами. Если сравнивать по этому параметру двумерную решетку и гиперкуб, то при увеличении числа узлов топология гиперкуба является более выгодной.

Время передачи информации от узла к узлу зависит от стартовой задержки и скорости передачи. Прогресс в производительности процессоров гораздо больше, чем в пропускной способности каналов связи. За время передачи процессорные узлы успевают выполнить большое количество команд. Поэтому инфраструктура каналов связи является одной из главных компонент кластерной или MPP-системы.

Благодаря маштабируемости именно кластерные системы являются сегодня лидерами по достигнутой производительности.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ ДЛЯ САМОПРОВЕРКИ

1. Дайте определение ЭВМ, вычислительной и информационной системам, архитектуре ЭВМ.
2. Опишите развитие и классификацию архитектур компьютеров.
3. Опишите конвейерную технологию выполнения команд.
4. Охарактеризуйте характерные черты суперскалярной обработки команд.
5. Приведите классификацию архитектуры SISD с характеристикой классов.
6. Определите основные характерные черты CISC-архитектуры.
7. Охарактеризуйте основные характерные черты RISC-архитектуры.
8. Укажите основные характерные черты VLIW-архитектуры.
9. Охарактеризуйте основные отличительные черты EPIC-концепции.
10. В чем суть матричного и векторно-конвейерного способов организации SIMD-архитектуры?
11. В чем суть MMX-технологии и потоковых SIMD-расширений?
12. Почему появились многоядерные структуры процессоров и технологии многопоточности?
13. Охарактеризуйте все виды производительности компьютера.
14. Как определить энергоэффективность процессора?
15. Приведите классификацию ЭВМ по назначению.
16. Приведите классификацию ЭВМ по функциональным возможностям.
17. Опишите функциональные возможности, области применения, современные разработки мэйнфреймов.
18. Опишите функциональные возможности, пути развития, современные разработки суперЭВМ.
19. Приведите классификацию микроЭВМ с краткой характеристикой классов.
20. Опишите функциональные возможности, назначение, платформы рабочих станций.
21. Приведите классификации серверов с пояснениями.

22. Перечислите требования, которые учитываются при проектировании серверов.

23. Охарактеризуйте класс модульных серверов.

24. Особенности многоузловых серверных систем.

25. Какими преимуществами обладают блейд-серверы?

26. Какими характеристиками должен обладать ПК?

27. Приведите классификацию ПК по способу использования и по назначению.

28. Дайте классификацию ноутбуков.

29. Охарактеризуйте планшетные ПК.

30. Приведите классификацию, состав, платформы карманных устройств.

31. Охарактеризуйте встраиваемые и промышленные компьютеры.

32. Опишите обобщенную структуру ЭВМ и пути ее развития.

33. Опишите типы данных в интеловских процессорах с IA-32, IA64.

34. Опишите типы данных в IA-64.

35. Опишите векторные типы данных.

36. Охарактеризуйте классические структуры команд.

37. Объясните суть ассоциативного и адресного поиска операндов в памяти.

38. Как осуществляется непосредственная, прямая и косвенная адресация операндов?

39. Каким образом реализуется адресация операндов «базирование способом суммирования» и «базирование способом совмещения»?

40. Как реализуется индексная адресация?

41. Как осуществляется адресация операндов «базирование с индексированием»?

42. Опишите развитие CISC-системы команд x86 (по годам).

43. Какие новые возможности появились у процессора с введением расширения команд SSE-2, SSE-3, SSE-4?

44. Опишите расширенеие AES-NI, AVX.

45. Опишите расширение AVX2, FMA3, BMI, MIC и SGX.

46. Охарактеризуйте особенности режимов работы процессоров AMD64, Intel64.

47. Опишите обобщенный формат команд x86 и форматы команд RISC-процессора.

48. Приведите формат команд IA-64 и структуру пакета инструкций.

49. Опишите принципы организации системы прерывания программ.

50. Приведите характеристики системы прерывания.

51. Как реализуется программно-управляемый приоритет прерывающих программ?

52. Определите назначение и структуру центрального процессора ЭВМ.

53. Приведите классификацию методов построения центрального устройства управления процессора.

54. Определите назначение, структуру, количество основных функциональных регистров IA-32 и регистров блока обработки чисел с плавающей точкой IA-32.

55. Определите назначение, структуру, количество регистров MMX-технологии и расширений SSE, SSE2.

56. Объясните суть процедуры переименования регистров в процессорах.

57. Опишите регистровые структуры процессоров AMD64, Intel64.

58. Опишите регистровые структуры процессоров IA-64.

59. Сформулируйте характерные черты современных универсальных микропроцессоров.

60. Опишите стратегию развития процессоров Intel.

61. Приведите особенности микроархитектуры процессоров IntelCore.

62. Приведите особенности микроархитектуры процессоров Intel Nehalem.

63. Как осуществляется декодирование команд x86 в процессоре Intel Nehalem?

64. Охарактеризуйте исполнительные устройства процессоров Intel Nehalem.

65. Приведите особенности процессора семейства Intel Westmere.

66. Приведите особенности процессорного ядра Sandy Bridge.

67. Охарактеризуйте кольцевую шину в Sandy Bridge?

68. Приведите особенности микроархитектуры процессоров Intel Haswell.

69. Опишите структуру графического ядра в Intel Haswell.

70. Приведите особенности микроархитектуры процессоров Intel Skylake.

71. Опишите конфигурацию процессоров Haswell и Skylake с eDRAM.

72. Опишите структуру графического ядра в Skylake.

73. Опишите историю создания и современное состояние семества микропроцессоров Эльбрус.

74. Особенности микроархитектуры IBM POWER8.

75. Опишите иерархическую структуру памяти компьютера.

76. Охарактеризуйте способы размещения данных в кэш-памяти.

77. Какие существуют методы обновления строк в основной и кэш-памяти?

78. Какие существуют методы замещения строк в кэш-памяти?

79. Опишите общие принципы организации оперативной памяти компьютера.

80. Охарактеризуйте способы распределения оперативной памяти.

81. Какие существуют методы повышения пропускной способности ОП?

82. Сформулируйте концепцию виртуальной памяти.

83. Опишите страничное и странично-сегментное распределение виртуальной памяти.

84. Перечислите характеристики интерфейсов.

85. Приведите классификацию интерфейсов.

86. Охарактеризуйте способы передачи данных в подсистеме ввода-вывода.

87. Опишите системную организацию на базе чипсетов компании Intel.

88. Приведите классификацию MIMD-систем по способу взаимодействия процессоров.

89. Охарактеризуйте сильносвязанные и слабосвязанные многопроцессорные системы.

СПИСОК ЛИТЕРАТУРЫ

1. Чередов А.Д. Организация ЭВМ и систем: учебное пособие / А.Д. Чередов; Томский политехнический университет. – 3-е изд., перераб. и доп. – Томск: Изд-во Томского политехнического университета, 2011. – 200 с.

2. Орлов С.А. Организация ЭВМ и систем: учебник для вузов / С.А. Орлов, Б.Я. Цилькер. – 3-е изд. – СПб.: Питер, 2014. – 688 с.

3. Новожилов О.П. Архитектура ЭВМ и системы. Учебное пособие для бакалавров. – М.: Изд-во Юрайт, 2015. – 527 с.

4. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации: учебное пособие для вузов / В.Л. Брайдо, О.П. Ильина. – 4-е изд. – СПб.: Питер, 2011. – 555 с.

Интернет-ресурсы

5. Официальный сайт компании Intel, США. – http:// www.intel.com
6. Официальный сайт компании AMD, США. – http:// www.amd.com
7. Официальный сайт компании HP, США. – http:// www.hp.com
8. Официальный сайт компании IBM, США. – http:// www.ibm.com
9. Сайт информационных технологий. – http:// www.ixbt.com
10. Сайт высоких технологий IT-индустрии. – http://citforum.ru

Учебное издание

ЧЕРЕДОВ Андрей Дмитриевич
МАЛЬЧУКОВ Андрей Николаевич

ОРГАНИЗАЦИЯ ЭВМ И СИСТЕМ

Учебное пособие

Научный редактор
доктор технических наук, профессор Н.Г. Марков

Редактор *Н.Т. Синельникова*

Верстка *Л.А. Егорова*

Отпечатано в Издательстве ТПУ в полном соответствии с качеством предоставленного оригинал-макета

Подписано к печати Формат 60×84/16.
Бумага «Снегурочка». Печать Xerox.
Усл. печ.л. 11,63. Уч.-изд. л. 10,53.
Заказ . Тираж 100 экз.

Национальный исследовательский
Томский политехнический университет
Система менеджмента качества
Издательства Томского политехнического университета сертифицирована
NATIONAL QUALITY ASSURANCE по стандарту BS EN ISO 9001:2008

ИЗДАТЕЛЬСТВО ТПУ. 634050, г. Томск, пр. Ленина, 30.
Тел./факс: 8(3822)56-35-35, www.tpu.ru