

Государственное образовательное учреждение
высшего профессионального образования
«Московский государственный университет леса»

Факультет электроники и системотехники
Кафедра электроники и микропроцессорной техники

И.В. Антошина, Ю.Т. Котов

Микропроцессоры и микропроцессорные системы

(аналитический обзор)



Москва 2005 г.

Рекомендовано к изданию редакционно-издательским советом
университета в качестве учебного пособия

Разработано в соответствии с Государственным образовательным стандартом ВПО 2000 г. На основе примерных программ дисциплин «Микропроцессорные системы» (спец. 220100 (230101) Вычислительные машины, комплексы, системы и сети), «Микропроцессорная техника в приборостроении» (спец. 551500 (200100) Приборостроение), «Микропроцессоры в системах управления».

Рецензент: доцент кафедры прикладной математики В.Г. Пугин

Пособие содержит сведения о принципах построения, особенностях функционирования современных микропроцессоров и микропроцессорных систем на их основе

От авторов

Исходные материалы для данной работы были заимствованы нами, в основном, из публикаций А. Андреева, А. Богданова, А. Власова, Д. Волкова, А. Галушкина, А. Геворкяна, А. Горбаня, Ю. Горбачева, В. Дьяконова, А. Дудкина, Е. Зудиловой, В. Корнеева, А. Кузнецова, М. Кузьминского, В. Мареева, Ю. Полякова, А. Пылкина, Д. Россиева, Е. Станковой, В. Степанова, А. Шадского, В. Шахнова, приведенных на различных сайтах в Internet. Мы не претендуем на новизну материалов в данной работе и не можем дать гарантии в полной достоверности полученных выводов ввиду очевидной неполноты исходного материала. Одновременно выражаем благодарность всем вышеперечисленным авторам за приведенные в Internet материалы и надеемся, что отсутствие ссылок по тексту на их работы не будет принято ими как нарушение нами их авторских прав.

Оглавление

	Стр.
Сокращения	5
Введение	6
1. Области применения микропроцессоров	7
2. Виды архитектур микропроцессоров	11
3. История развития микропроцессоров	
3.1. Микропроцессоры первого поколения	16
3.2. Микропроцессоры второго поколения	27
3.3. Микропроцессоры третьего поколения	30
3.4. Микропроцессоры четвертого поколения	32
3.5. Микропроцессоры пятого поколения	34
3.6. Микропроцессоры шестого поколения	36
4. Классификация микропроцессоров	51
5. Особенности архитектуры 32-разрядных МП	
5.1. Микропроцессоры с RISC –архитектурой	
5.1.1. Общие принципы построения	61
5.1.2. Берклийская архитектура	68
5.1.3. Станфордская архитектура	71
5.1.4. Применение RISC- архитектуры в 32-разрядных МП	74
5.1.5. Особенности интеграции элементов RISC- архитектуры в процессорах серии x86	78
5.2. МП с традиционной RISC - архитектурой	
5.2.1. Intel Pentium 4	82
5.2.2. AMD Athlon	96
5.2.3. MC88110 компании Motorola	99
5.3. Микропроцессоры с масштабируемой архитектурой	
5.3.1. SuperSPARC	103
5.3.2. MicroSPARC-II	106
6. Особенности архитектуры 64 – разрядных МП	
6.1. Itanium 2 Intel	108
6.2. Athlon 64 AMD	112
6.3. UltraSPARC 111 Sun	117
6.4. Alpha 21264 DEC	127
6.5. PA 7100 Hewlett-Packard	135
6.6. R12000 MIPS	143
6.7. PowerPC 970 IBM	146
7. Микропроцессоры нетрадиционных архитектур	

7.1	Ассоциативные процессоры	149
7.2.	Матричные процессоры	150
7.3.	ДНК процессоры	152
7.4.	Клеточные процессоры	155
7.5.	Коммуникационные процессоры	157
7.6.	Процессоры баз данных	159
7.7.	Потоковые процессоры	160
7.8.	Процессоры с многозначной (нечеткой) логикой	162
7.9.	Сигнальные процессоры	164
8.	Архитектуры микропроцессорных систем	
8.1	Структуры с централизованным, децентрализованным и комбинированным управлением	174
8.2.	Системы с перестраиваемой структурой	176
8.3.	Системы с резервированием	178
8.4.	Иерархические системы	179
8.5.	Однопроцессорная МПС типа «Общая шина»	183
8.6.	Архитектуры с параллельной обработкой данных	186
8.6.1.	SMP архитектура	190
8.6.2.	MPP архитектура	192
8.6.3.	Гибридная архитектура	193
8.6.4.	PVP архитектура	195
8.6.5.	Кластерная архитектура	196
8.6.6.	Транспьютеры	202
8.6.7.	MBC – 1000	215
8.6.8.	Молекулярные МПС	218
8.6.9.	Оптические МПС	225
8.6.10.	Нейронная архитектура	243
8.6.11.	Масштабируемая архитектура	260
9.	Многопроцессорные системы	
9.1.	Общие требования, предъявляемые к МПС	263
9.2.	Классификация систем параллельной обработки данных	268
9.3.	Модели связи и архитектуры памяти	276
9.4.	Многопроцессорные системы с общей памятью	279
9.5.	Многопроцессорные системы с локальной памятью	290
10.	Режимы обмена в МПС	294
11.	Каналы передачи информации в МПС	310
12.	Организация памяти МПС	332
13.	Технологические аспекты полупроводниковой технологии	350

Приложение 1 История развития компьютеров - период до появления первого ПК	360
Приложение 2 История развития МП Intel	368
Приложение 3. Сводные данные о МП Intel	400
Приложение 4. Первая десятка самых мощных микропроцессорных систем (суперкомпьютеров)	430

Сокращения:

ПК – персональный компьютер
МП - микропроцессор
МПС – микропроцессорная система
УВВ – устройство ввода - вывода
АЛУ – арифметико-логическое устройство
ПО – программное обеспечение
ВУ – внешнее устройство
ВС – вычислительная система
БИС (СБИС) – большая (сверхбольшая) интегральная схема
СОЗУ – сверхбыстрое оперативное запоминающее устройство
ЦПЭ – центральный процессорный элемент
ОЗУ – оперативное запоминающее устройство
ЗУ – запоминающее устройство
ЭВМ – электронная вычислительная машина
РОН – регистр общего назначения
ШД – шина данных
ЦП – центральный процессор
ША – шина адреса
ПЗУ – постоянное запоминающее устройство
АЛУ – арифметико – логическое устройство
ПДП – прямой доступ к памяти
ОС – операционная система
МПК БИС – микропроцессорный комплект БИС
ЦПЭ – центральный процессорный элемент
КВВ – контроллер ввода - вывода
КПр – контроллер прерываний
ИНС – искусственная нейронная сеть
ССП – слово состояния процессора

Введение

Бурные темпы развития цифровых методов обработки информации влекут разработку и всеобщее внедрение в практику вычислений и управления производством микропроцессорных средств (персональных компьютеров и соответствующего периферийного оборудования) обработки информации. Их аппаратурная реализация, включающая микропроцессоры, контроллеры, системные платы, шины, накопители, системы вывода видео- и аудиоинформации и т.д., во многом обеспечивает заданные уровни вычислительной мощности и функциональных возможностей систем, использующих эти средства.

Разработкой аппаратурных средств занимаются множество фирм. Использование при разработке различных конструктивно-технологических принципов порождает большое разнообразие вариантов их построения для построения высокопроизводительных вычислительных систем (ВС), основу которых составляют микропроцессорные средства обработки информации (микропроцессорные системы – МПС).

При разработке МПС основным практически всегда стоит вопрос выбора оптимального состава аппаратурных средств и, прежде всего, микропроцессоров (МП), обеспечивающих получение максимально возможной эффективности работы системы. В реальных условиях поиск соответствующих материалов для решения данного вопроса почти всегда оставался сложным из-за недостаточности публикаций обобщающего характера. В данной работе сделана попытка провести обзор наиболее известных существующих микропроцессоров и общих принципов построения систем на их основе.



1. Области применения микропроцессоров

При построении различных микропроцессорных систем учету подлежат различные технические и производственно-технологические факторы, влияющие на эффективность использования систем в аппаратуре. Состав аппаратуры МПС должен обеспечивать:

- простое наращивание разрядности и производительности,
- возможность широкого распараллеливания вычислительного процесса,
- эффективную обработку алгоритмов решения различных задач,
- простоту технической и математической эксплуатации.

Сама МПС, будучи оснащенной разнообразными устройствами ввода - вывода (УВВ) информации, может применяться в качестве законченного изделия. Однако часто к МПС необходимо подавать сигналы от множества измерительных датчиков и исполнительных механизмов какого - либо сложного объекта управления или технологического процесса. В этом случае уже образуется сложная вычислительная система, центром которой является МП.

Простые в архитектурном исполнении микропроцессоры применяются для измерения временных интервалов, управления простейшими вычислительными операциями (в калькуляторах), работой кино-, фото-, радио- и телеаппаратуры. Они используются в системах охранной и звуковой сигнализации, приборах и устройствах бытового назначения. Бурно развивается производство электронных игр с использованием микропроцессоров. Они порождают не только интересные средства развлечения, но и дают возможность проверять и развивать приемы логических заключений, ловкость и скорость реакции.

Видеоигры можно отнести к приложениям, требующим использования компьютеров с ограниченным набором функций. Сегодня игровые приставки потребляют наибольшее количество, если не считать ПК, 32 - разрядных микропроцессоров. Наибольшее применение здесь получили МП Intel, Motorola. В устройстве PlayStation фирмы Sony используется 32 - разрядный процессор MIPS, а в видеоприставке Nintendo 64 — даже 64 - разрядный чип

того же производителя. Продукты компании Sega с видеоиграми Saturn и Genesis вывели RISC - процессоры серии SH фирмы Hitachi на третье место в мире по объему продаж среди 32 - разрядных систем.

Хорошие перспективы сулит 32 - разрядным процессорам рынок персональных электронных секретарей (PDA) и электронных органайзеров. Современные электронные органайзеры - яркий пример интегрированных приложений, ведь для них практически не существует независимых поставщиков программного обеспечения. С другой стороны, PDA типа Newton фирмы Apple, по сути, не что иное, как новая вычислительная платформа, будущее которой зависит от разработчиков программного обеспечения (ПО).

До настоящего времени успехом среди электронных органайзеров пользуются устройства с ограниченным набором функций. Тем не менее, дальнейшее совершенствование технологии может вывести эти «ручные» компьютеры в абсолютные лидеры, которые по объемам продаж в натуральном выражении должны обойти ПК.

Важной функцией МП является предварительная обработка информации с внешних устройств (ВУ), преобразования форматов данных, контроллеров электромеханических внешних устройств. В аппаратуре МП дает возможность производить контроль ошибок, кодирование - декодирование информации и управлять приемо-передающими устройствами. Их применение позволяет в несколько раз сократить необходимую ширину телевизионного и телефонного каналов, создать новое поколение оборудования связи.

Использование МП в контрольно-измерительных приборах и в качестве контрольных средств радиоэлектронных систем дает возможность проводить калибровку, испытание и поверку приборов, коррекцию и температурную компенсацию, контроль и управление измерительными комплексами, преобразование и обработку, индикацию и представление данных, диагностику и локализацию неисправностей.

С помощью микропроцессорных средств можно решать сложные технические задачи по разработке различных систем сбо-

ра и обработки информации, где общие функции сводятся к передаче множества сигналов в один центр для оценки и принятия решения. Например, в бортовых системах летательных аппаратов за время полета накапливается большое количество информации от различных источников, требующих зачастую незамедлительной ее обработки. Это осуществляется централизованно с помощью вычислительной системы на основе бортовой МПС.

Обобщая рассмотренные примеры использования МП, можно выделить четыре основных направления их применения:

- встроенные системы контроля и управления;
- локальные системы накопления и обработки информации;
- распределенные системы управления сложными объектами,
- распределенные высокопроизводительные системы параллельных вычислений.

Встроенные системы контроля и управления. Управляющие встроенные МПС предназначены для решения локальных задач управления объектами и могут выполнять функции контроллеров устройств, подключаемых к МПС более высоких контуров управления или быть центром управляющих систем нижних контуров управления.

Использование МПС даже в простейшей схеме управления принципиально изменяет качество функционирования обслуживаемых им устройств. Она позволяет оптимизировать режимы работы управляемых объектов или процессов и за счет этого получать прямой и/или косвенный технико-экономический эффект.

Прямой технико-экономический эффект выражается в экономии потребляемой энергии, повышении срока службы и снижении расхода материалов и оборудования. Косвенный технико-экономический эффект связан со снижением требований к обслуживающему персоналу и повышением производительности.

Опыт показывает, что практически во всех случаях использование МПС только за счет экономии электроэнергии обеспечивается ее окупаемость за 1 - 1.5 года. Управление оборудованием на основе встроенных систем контроля и управления создает реальные предпосылки создания полностью автоматизированных производств.

Использование МПС повышает качество работы и произво-

длительность оборудования, существенно снижает требования к персоналу, работающему на нем. Цифровое управление отдельными единицами оборудования на различных уровнях позволяет легко собирать информацию (или вызвать ее) с нижних на верхние уровни иерархической системы управления.

Локальные системы накопления и обработки информации.

Уровень управления современным предприятием или учреждением требует наличия для любого специалиста или руководителя достаточно большого объема специфичной информации. Это может быть обеспечено за счет применения локальных микропроцессорных вычислительных систем.

Локальные, т. е. расположенные на рабочем месте, МПС накопления и обработки информации экономически и технически просто осуществляют информационное обеспечение потребителей. Объединение локальных систем между собой в сеть и дистанционное подключение этой сети к центральной ЭВМ с громадным информационным архивом позволяют создать завершённую автоматизированную систему информационного обеспечения.

Внешние устройства локальных МПС могут встраиваться в корпус ЭВМ. Их устройства образуют комплект, минимально необходимый для проведения вычислительных работ и обработки данных. В комплект сложных локальных МПС, ориентированных на решение инженерных и научных задач, могут входить разнообразные внешние устройства, например, печати, визуального отображения, внешней памяти, комплексирования, пульта операторов общего назначения и т. д.

Распределенные системы управления сложными объектами.

Альтернативой широко распространенным системам с центральным процессором становятся распределенные микропроцессорные управляющие системы. В этом случае микропроцессоры и связанные с ними схемы обработки данных физически располагаются вблизи мест возникновения информации, образуя локальные МПС. Такое построение системы позволяет вести обработку информации на месте ее возникновения, например, вблизи двигателей, рулей управления, тормозной системы и т. д. В этом случае связь системы с центральной системой обработки и накопления данных и создает пространственно - распределенную систему управления.

В распределенных системах достигается значительный рост быстродействия получения и обработки входной информации, экономия в количестве и распределении линий связи, повышается живучесть, существенно развиваются возможности оптимизации режимов управления и функционирования.

Распределенные высокопроизводительные системы параллельных вычислений. МПС открыли новые возможности решения сложных вычислительных задач, алгоритмы вычисления которых допускают распараллеливание, т. е. одновременные (параллельные) вычисления на многих микропроцессорах.

Системы параллельных вычислений на основе десятков, сотен и даже тысяч одинаковых или специализированных на определенные задачи микропроцессоров при значительно меньших затратах дают такую же производительность, как и вычислительных системах на основе мощных процессоров конвейерного типа. Создание МПС с большим количеством специализированных по функциональному назначению процессоров позволяет проектировать мощные ВС нового типа по сравнению с традиционными развитыми большими вычислительными системами.

2. Виды архитектур микропроцессоров

Термин «архитектура» носит двойной смысл. В первом случае под архитектурой понимается архитектура набора команд, исполняемых микропроцессором. Во втором случае архитектура охватывает понятие организации системы, включающее структуру памяти, системной шины, организацию ввода/вывода и т.п. Применительно к вычислительным системам термин «архитектура» может быть определен как распределение функций, реализуемых системой, между ее уровнями.

Так, например, архитектура первого уровня определяет, какие функции по обработке данных выполняются МП в целом, а какие возлагаются на внешний мир (пользователей, операторов, администраторов баз данных и т.д.). МП взаимодействует с внешним миром через набор интерфейсов: языков (оператора, программирования, описания, манипулирования базой данных,

управления заданиями) и системных программ (служебных, редактирования, сортировки, сохранения и восстановления информации).

Архитектура второго уровня может разграничивать определенные уровни внутри программного обеспечения. Например, уровень управления логическими ресурсами может включать реализацию таких функций, как управление базой данных, файлами, виртуальной памятью, сетевой телеобработкой. К уровню управления физическими ресурсами относятся функции управления внешней и оперативной памятью, управления процессами, выполняющимися в системе.

Следующий, третий, уровень отражает основную линию разграничения системы, а именно границу между системным программным обеспечением и аппаратурой. Эту идею можно развить и дальше и говорить о распределении функций между отдельными частями физической системы. Например, некоторый интерфейс определяет, какие функции реализуют центральные процессоры, а какие - процессоры ввода/вывода.

Архитектура четвертого уровня определяет разграничение функций между процессорами ввода/вывода и контроллерами внешних устройств. В свою очередь можно разграничить функции, реализуемые контроллерами и самими устройствами ввода/вывода (терминалами, модемами, накопителями на магнитных дисках и лентах). Архитектура таких уровней часто называется архитектурой физического ввода/вывода.

При создании МП используются три наиболее широко применяемых вида архитектур, созданных за время их развития: регистровая, стековая и ориентированная на оперативную память.

Регистровая архитектура (архитектура типа «регистр - регистр») микропроцессора определяет наличие достаточно большого набора регистров внутри больших интегральных схем (БИС) микропроцессора. Этот набор регистров образует поле сверхбыстрой оперативной памяти (СОЗУ) с произвольной записью и выборкой информации.

В микропроцессорах с регистровой архитектурой рабочие области регистров размещаются в логических частях процессоров. Однако малая плотность логических схем по сравнению с плотно-

стью схем памяти ограничивает возможность регистровой архитектуры. МП с архитектурой, ориентированной на память, обеспечивают быстрое подключение к рабочим областям, когда необходимо заменять контексты. Смена контекстов осуществляется изменением векторов трех регистров - счетчика команд, регистров состояния и указателя рабочей области. Достоинство этой архитектуры в отношении смены контекстов связано с выполнением только одной команды для передачи полного вектора контекста.

Микропроцессоры с регистровой архитектурой имеют высокую эффективность решения научно - технических задач, поскольку высокая скорость работы СОЗУ позволяет эффективно использовать скоростные возможности арифметик - логического блока. Однако при переходе к решению задач управления эффективность таких микропроцессоров падает, так как при переключениях программ необходимо разгружать и загружать регистры СОЗУ.

Стековая архитектура микропроцессора дает возможность создать поле памяти с упорядоченной последовательностью записи и выборки информации. Эта архитектура эффективна для организации работы с подпрограммами, когда возникает постоянная необходимость перехода от текущей программы к подпрограмме, обслуживающей какое - либо ВУ, и возврат в текущую программу. Хранение адресов возврата позволяет организовать в стеке эффективную обработку последовательностей вложенных подпрограмм.

Основным недостатком МП этого типа является то, что стек, реализованный на кристалле микропроцессора, как правило имеет малую информационную емкость. При работе он быстро переполняется, приводя к возможности нарушения работы системы. Построение же стека большой емкости требует значительных ресурсов кристалла. Поэтому наилучшими характеристиками обладают МП, в которых стек реализуется вне микропроцессора - в оперативной памяти (оперативном запоминающем устройстве - ОЗУ).

Архитектура микропроцессора, ориентированная на оперативную память, обеспечивает высокую скорость работы и большую информационную емкость рабочих регистров и стека

при их организации в ОЗУ. В МП с такой архитектурой все обрабатываемые числа после операции в микропроцессоре выводятся из микропроцессора и вновь возвращаются в память, что и дало ей такое название.

При оценке быстродействия МП типа «память - память» необходимо учитывать физическую реализацию как элементов, так и связей между ними. Высокая скорость срабатывания логических элементов интегральных схем не всегда может обеспечить высокую скорость работы МП, поскольку большие значения индуктивно - емкостных параметров связей на печатных платах не позволяют передавать сигналы без искажения. Высокий уровень технологии современных МП до долей микрон существенно уменьшило размеры БИС, снизило паразитные параметры связей. Поэтому стало возможным физически отделить блок регистров и стек от арифметико-логического блока и обеспечить при этом их высокоскоростную совместную работу. При создании однокристалльных МП регистровые СОЗУ и ОЗУ МПС имеют практически одни и те же параметры. Повышение скорости работы ОЗУ позволяет удалить набор регистров и стек из кристалла микропроцессора и использовать освободившиеся ресурсы для развития системы команд, средств прерывания, многозарядной обработки. Организация рабочих регистров и стека в ОЗУ ведет к уменьшению скорости передачи информации, однако при этом повышается общая эффективность такого решения за счет большой информационной емкости полей регистровой и стековой памяти, а также возможности развития системы команд и прерываний.

Архитектура микропроцессора, ориентированная на оперативную память, обеспечивает экономию площади кристалла МП. В этом случае на кристалле размещается только регистр - указатель начального файла набора регистров. Адресация остальных регистров осуществляется указанием в команде специальным указателем - кодом смещения. Доступ к рабочим регистрам в этом случае замедляется, поскольку приходится совершать сопряженное с затратами времени кольцевое «путешествие» из процессора во внекристалльную память, где размещаются рабочие регистры. Однако контекстное переключение в микропроцессоре с такой архитектурой происходит быстро, поскольку при прерывании необхо-

димо только изменить значение содержимого регистра - указателя рабочей области памяти.

Другая отличительная особенность архитектуры МП, ориентированной на оперативную память - двухадресный формат команд. В этих МП нет специального накапливающего регистра, выполняющего функции подразумеваемой ячейки результата для всех двухоперандных команд. Результат формируется в соответствии с алгоритмом, приведенном для примера на рис. 1,а, где операция сложения содержимого двух ячеек памяти с номерами X и Y осуществляется по команде «сложить XY».

Поскольку в архитектуре типа «память - память» любая ячейка памяти может содержать либо исходный операнд, либо операнд-результат, то эта операция выполняется по одной команде. В то же время в процессорах с одноадресной регистровой архитектурой для достижения той же самой цели приходится использовать две команды:

- команду пересылки операнда Y во внутренний регистр Rг,
- команду сложения содержимого внутреннего регистра Rг с содержимым ячейки памяти X и пересылки результата в ячейку X (рис. 1,б).

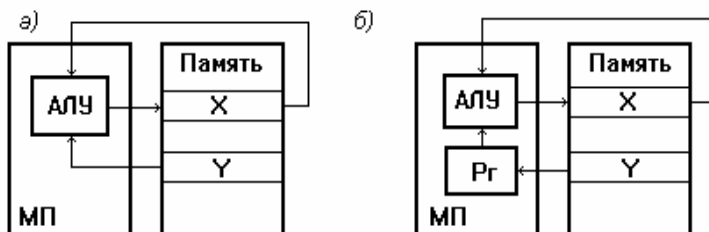


Рис. 1

В первом случае при компиляции программ для компиляторов высокоуровневых языков существенно упрощается задача присвоения значений переменным и, благодаря этому, получаются более короткие модули объектных программ.

Использование возможностей быстрой смены контекстов и фактически неограниченной рабочей области в МП с архитектурой, ориентированной на оперативную память, позволяет им легко

находить применение в МПС, работающим в реальном масштабе времени.

К достоинствам архитектуры МП, ориентированной на оперативную память, относится возможность развития системы, позволяющая снизить время разработки ПО. Здесь под развитием понимается способность систем внедрять в виде функциональных модулей программные, программно - аппаратурные и даже аппаратурные средства, которые можно использовать в системе по мере совершенствования аппаратурных средств и накопления опыта.

Распределенные системы управления часто требуют применения полуавтономных контроллеров, которые должны вписываться в определенные иерархические структуры. При этом архитектура МП, ориентированная на память, обеспечивает естественный и эффективный интерфейс между контроллерами, расположенными на одном иерархическом уровне, и процессорами управления, расположенными на более высоком иерархическом уровне, а структура связей между контроллерами может быть обеспечена за счет развитых информационных магистралей.

3. История развития микропроцессоров

3.1. Микропроцессоры первого поколения

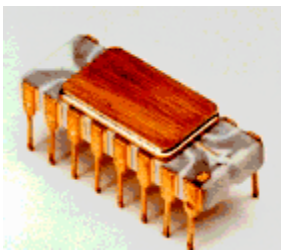
Микропроцессор обязан своему появлению внедрению в начале 70-х годов в производство микроэлектронной элементной базы, основу которой составляют интегральные схемы (ИС). Разработкой и производством МП в то время занимались ряд фирм США, среди которых наиболее совершенной технологией на фоне других обладала фирма Intel. Поэтому историю развития рассмотрим на примере развития МП данной фирмы.

До начала 70-х годов вычислительные машины были доступны весьма ограниченному кругу специалистов, а их применение, как правило, оставалось окутанным завесой секретности и мало известным широкой публике. Однако в 1971 г. произошло событие, которое резко изменило ситуацию и превратило вычислительную машину в повседневный рабочий инструмент десятков миллионов людей.

В 1971 году фирма Intel из небольшого американского городка Санта-Клара (шт. Калифорния) создала новый полупроводниковый прибор, получивший название «микропроцессор». В 1968 г. Гордон Мур и Боб Нойс, одни из тех, кто закладывал фундамент известной полупроводниковой компании Fairchild Semiconductor, основали фирму Intel Corporation. Первой идеей нового предприятия было создание полупроводниковых запоминающих устройств, призванных заменить ЗУ на магнитных сердечниках. Поскольку к концу 60-х годов память этого типа практически исчерпала весь свой потенциал развития, проблема была весьма актуальной, а ее разработка сулила немалые прибыли. И хотя в данной области Intel добилась заметных успехов, тем не менее мировую славу ей принесли совсем другие изделия.

Поворотным моментом в истории компании стал 1969 г., когда был получен заказ на создание ряда специализированных микросхем для калькуляторов от ныне уже несуществующей японской фирмы Busicom. В результате его реализации был разработан кристалл в сопровождении соответствующих средств поддержки.

МП i4004. 15 ноября 1971 г. Intel приступила к поставкам первого в мире микропроцессора Intel 4004 - именно такое обозначение получил первый прибор, послуживший отправной точкой абсолютно новому классу полупроводниковых устройств.



Кристалл представлял собой 4-разрядный процессор и изготавливался по передовой в те годы р-канальной МОП-технологии с проектными нормами 10 мкм. Электрическая схема прибора насчитывала 2300 транзисторов. Микропроцессор работал на тактовой частоте 750 кГц при длительности цикла команды 10,8 мкс.

МП i4004 имел адресный стек (счетчик команд и три регистра стека типа LIFO - Last In First Out), блок регистров общего назначения - РОН (регистры сверхоперативной памяти, или регистровый файл), 4-разрядное параллельное АЛУ, аккумулятор, регистр команд с дешифратором команд и схемой управления, а также схему связи с периферийными устройствами. Все эти функ-

циональные узлы объединялись между собой 4-разрядной шиной данных (ШД).

Память команд достигала 4Кбайт (для сравнения: объем ЗУ мини-ЭВМ в начале 70-х годов редко превышал 16 Кбайт), а регистровый файл центрального процессора (ЦП) насчитывал шестнадцать 4-разрядных регистров, которые можно было использовать и как восемь 8-разрядных (восемь 4-разрядных пар). Такая организация РОН сохранена и в последующих микропроцессорах фирмы Intel. Три регистра стека обеспечивали три уровня вложения подпрограмм.

МП i4004 монтировался в пластмассовый или металлокерамический корпус типа DIP (Dual In-line Package) всего с 16 выводами.

В систему его команд входило 46 инструкций. По своему функциональному составу она была универсальной, т. е. рассчитана на широкий круг решаемых задач и разрабатываемых приложений. Первоначальное назначение кристалла наложило определенный отпечаток на состав системы команд, поэтому присутствие в ней ряда инструкций, в частности десятичной коррекции, а также наличие соответствующих аппаратных средств не вызывает особого удивления.

Вместе с тем кристалл располагал весьма ограниченными средствами ввода/вывода, а в системе команд отсутствовали операции логической обработки данных (И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ), в связи с чем их приходилось реализовывать с помощью специальных подпрограмм, что в некоторых случаях чрезмерно усложняло создаваемое ПО.

МП i4004 не имел возможности останова (команды HALT) и обработки прерываний. Это объясняется тем, что в калькуляторах, где поначалу и планировалось использовать прибор, особой необходимости в этих средствах нет.

Цикл команды процессора состоял из восьми тактов задающего генератора. Как уже отмечалось, МП i4004 монтировался в корпус всего с 16 выводами - самый распространенный (а значит, и самый дешевый) тип корпуса в начале 70-х годов. А поскольку в распоряжении инженеров оказался узкий интерфейс с "внешним миром", то пришлось пойти на применение мультиплексированной

шины адреса (ША) и данных, причем 12 - разрядный адрес выдавать порциями по четыре разряда, что, конечно, не могло не сказаться на длительности машинного цикла. Прием команды по такому интерфейсу требовал еще двух тактов. На исполнение же самой инструкции из восьми тактов процессор затрачивал лишь три.

Компанией было разработано и выпущено целое семейство БИС, в которое вошли постоянное запоминающее устройство (ПЗУ) 4001, ОЗУ 4002, регистр сдвига 4003 и ряд других вспомогательных микросхем. Поскольку все они были рассчитаны на совместное использование, разработка аппаратных средств системы заметно упрощалась.

Опыт использования первого МП показал, что такие факторы, как отсутствие средств обработки прерываний, наличие трех уровней вложения подпрограмм и необходимость реализации логических операций И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ с помощью специальных подпрограмм, далеко не всегда удовлетворяют разработчиков. Указанные недостатки привели к созданию нового МП i4040.

МП i4040. В i4040 сохранены все функциональные возможности предшествующей модели и существенно улучшены как технические, так и программные средства. Система команд пополнилась 14 инструкциями, включая выполнение логических операций И и ИЛИ; кроме того, в процессор были введены средства останова и обработки прерываний.

Претерпела некоторые изменения и архитектура устройства. Адресный стек процессора увеличен с трех до семи регистров, а количество РОН возросло с 16 до 24, причем их разбили на две области, выбираемые при помощи специальных команд. Отчасти такая организация обусловлена тем, что процессор теперь мог обращаться к двум блокам памяти команд объемом 4 Кбайт и за каждым из них программист мог закрепить свою область регистров. Наряду с этим восемь РОН были всегда доступны для использования. В итоге получилась достаточно гибкая и удобная структура, позволявшая разрабатывать самостоятельные программные модули, способные взаимодействовать через общую часть регистрового файла.

Обработка одноуровневых прерываний является одним из наиболее существенных новшеств МП i4040. Эта функция превратила его в полноценный процессор и сделала возможным использование в системах реального масштаба времени. Благодаря применению сигнала «останов» стала реальностью синхронизация работы процессора с некоторыми внешними событиями.

Несмотря на то что тактовая частота и машинный цикл i4040 не претерпели изменений, производительность МП возросла за счет использования более совершенной архитектуры и эффективной системы команд. 60 инструкций, ориентированных на широкий спектр решаемых задач, обработка прерываний, до 8 Кбайт памяти команд, а также возможность быстрого перевода систем на базе i4004 на новый процессор вывели i4040 в безусловные лидеры рынка 4-разрядных устройств.

МП i8008. Intel с 1 апреля 1972 г. начала поставки первого в отрасли 8-разрядного прибора Intel 8008. Он был разработан для нужд американской фирмы Computer Terminals Corporation of Texas, позднее известной как Datapoint.

Проектирование i8008 шло практически параллельно с работами над i4004. Кристалл изготавливался по р-канальной МОП-технологии с проектными нормами 10 мкм и содержал 3500 транзисторов. Процессор работал на частоте 500кГц при длительности машинного цикла 20 мкс (10 периодов задающего генератора).

В отличие от своих предшественников новый МП допускал применение комбинации ПЗУ и ОЗУ. Помимо увеличения разрядности и перехода на использование общего поля памяти для команд и данных, структура процессора претерпела еще ряд существенных изменений. Прежде всего это коснулось регистрового файла и устройства управления. По сравнению с i4004 число РОН уменьшилось вдвое (с 16 до 8), причем два регистра в основном использовались для хранения адреса при косвенной адресации памяти.

В связи с этим следовало бы ожидать снижения производительности, которого на самом деле не произошло, поскольку операции с памятью i8008 выполнял быстрее предыдущих моделей благодаря меньшему количеству состояний в машинном цикле и отсутствию необходимости исполнения минимум трех подготови-

тельных команд (как в i4004 и i4040) при обращении к ОЗУ или ПЗУ.

Вместе с тем объем блока регистров был ограничен возможностями технологии, которая в то время еще не позволяла размещать на кристалле большие регистровые структуры (в МП i8008 блок РОН был реализован в виде динамической памяти).

Почти вдвое (с восьми до пяти состояний) сократилась длительность машинного цикла. Теперь процессор выполнял команды за один - три машинных цикла, а некоторые инструкции - за один цикл из трех состояний. Для синхронизации работы МП с медленными устройствами был введен сигнал готовности (READY).

Разработчики технических средств на базе i8008 не были ограничены жесткими требованиями в отношении быстродействия микросхем памяти и периферийных устройств и могли использовать те ИС, которые наиболее полно соответствовали конкретной системе. В ряде случаев это приводило к ощутимому сокращению стоимости оборудования.

Система команд первого 8-разрядного МП насчитывала 65 инструкций, причем значительно увеличилось число команд условных переходов, а также логических инструкций и команд сдвига. Новый кристалл мог адресовать память объемом до 16 Кбайт (объем ЗУ для МП типа i4040 не превышал 8 Кбайт). Его производительность по сравнению с 4-разрядными системами возросла в 2,3 раза.

Процессор с такими параметрами уже можно было применять для построения контрольно - испытательного оборудования, прецизионной измерительной техники и сложных промышленных контроллеров систем управления технологическими процессами.

Однако i8008 имел свои недостатки. Объем и организация стека остались такими же, как и у i4040, и реализация операций с ним по - прежнему возлагалась на программиста. Узкий интерфейс с "внешним миром" ограничил количество управляющих сигналов процессора: в результате специалистам Intel пришлось использовать их шифрацию, что повлекло за собой необходимость установки дополнительного внешнего оборудования для формирования сигналов управления. В среднем для сопряжения процессора с па-

мятью и устройствами ввода/вывода требовалось около 20 схем средней степени интеграции.

Вскоре после выхода i8008 появилась его усовершенствованная версия i8008-1. Модернизированный вариант работал уже на частоте 800 кГц при длительности машинного цикла 12,5 мкс. Увеличение в 1,5 раза производительности центрального процессора наряду с большим (по тому времени) объемом оперативной памяти послужило лучшей рекомендацией для активного использования кристалла в различных областях, начиная от промышленности и медицины и кончая военной электроникой и торговлей. По мере расширения сферы влияния МП и усложнения систем на его базе возросли и требования к нему со стороны проектировщиков оборудования.

Несмотря на значительный успех разработанного кристалла среди проектировщиков систем, ПО к этому времени уже с трудом вписывалось в 16 Кбайт, да и производительность прибора начинала не удовлетворять многих разработчиков. Кроме того, некоторые области применения настойчиво требовали расширения не только количества, но и номенклатуры периферийных устройств. Системщики уже с трудом могли обходиться без такой традиционной для мэйнфреймов и мини-ЭВМ периферии, как дисплеи, принтеры, накопители на магнитной ленте и дисках и т. п. Стало очевидно, что технические характеристики изделия превратились в фактор, сдерживающий его дальнейшее распространение.

Возможности r-канальной МОП-технологии для создания сложных высокопроизводительных МП были уже практически исчерпаны, поэтому направление главного удара перенесли на технологию n-МОП. Перед проектировщиками стояли не менее сложные проблемы - разработка эффективной системы команд, рассчитанной на широкий круг решаемых задач, при сохранении программной совместимости с предыдущей моделью, расширение объема адресуемой памяти, поддержка интенсивного ввода/вывода без существенной потери производительности процессора, совершенствование подсистемы обработки прерываний. Указанные причины привели к созданию нового прибора – i8080.

МП i8080 - триумф 8-разрядных систем, который появился 1 апреля 1974 г. Благодаря использованию технологии n-МОП с

проектными нормами 6 мкм, на кристалле удалось разместить 6 тыс. транзисторов. При этом геометрические размеры самого кристалла по сравнению с i8008 увеличились незначительно. Следовательно, процент выхода годных изделий и ряд экономических показателей производства, включая себестоимость, удалось сохранить на достаточно высоком уровне. Тактовая частота процессора была доведена до 2 МГц, что в 2,5 раза превышало аналогичный параметр для i8008, а длительность цикла команды составила уже 2 мкс.

Несмотря на чисто внешнее сходство структур i8080 и i8008, схема нового процессора существенно отличалась от предшествующей модели. Объем памяти, адресуемой процессором, был увеличен в четыре раза и достиг 64 Кбайт (кстати, в то время ОЗУ такой емкости предлагали потребителям минимальные конфигурации многих мини-ЭВМ). В сочетании с эффективным механизмом обработки прерываний это давало им возможность широкого применения нового МП в сложных системах сбора и обработки информации различного назначения, особенно функционирующих в реальном масштабе времени. За счет использования корпуса с 40 выводами удалось разделить адресную и информационную шины процессора, в результате отпала необходимость применения дополнительных внешних схем для разделения потоков адресов и данных. Общее же количество микросхем, требовавшихся для построения системы в минимальной конфигурации, сократилось с 20 до 6, т. е. более чем в три раза. В регистровый файл были введены указатель стека, активно используемый при обработке прерываний, а также два программно-недоступных регистра для внутренних пересылок.

Поскольку предыдущий МП i8008 имел большой успех и для был наработан достаточно большой объем ПО, то сохранение разработчиками программной совместимости i8080 и i8008 было вполне естественным и разумным шагом. Именно поэтому в состав РОН нового процессора были включены основные рабочие регистры предыдущей модели. Правда, полной совместимости с i8008 достичь не удалось, так как процедуры обращения к подпрограммам и инструкции ввода/вывода МП i8080 в значительной степени отличались от соответствующих процедур и операций

кристалла i8008, и при переводе систем со старого процессора на новый в некоторых случаях программы приходилось полностью перерабатывать.

Включение в систему команд ряда инструкций, адресующих память с использованием трех пар регистров (в i8008 для этого выделялась одна пара), придало дополнительную гибкость. Реализация же блока РОН на основе статической, а не динамической памяти дала дополнительную экономию площади кристалла для размещения других схем процессора. Исключение аккумулятора из регистрового файла и введение его в состав арифметико - логического устройства упростило схему управления внутренней шиной, поскольку при этом отпала необходимость в ее использовании для передачи данных между сверхоперативной памятью и арифметико-логическим устройством (АЛУ) во время выполнения арифметических и логических операций.

Новым веянием в архитектуре МП стало использование многоуровневой системы прерываний по вектору. Такое техническое решение позволило довести общее число источников прерываний в системе до 256. Правда, до появления специализированных БИС контроллеров прерываний схема формирования векторов прерываний требовала применения до десяти дополнительных чипов средней степени интеграции.

Освобождение центрального процессора от управления ВУ и обмен данными между памятью системы и периферией, минуя ЦП, были уже достаточно давно и успешно реализованы в универсальных ЭВМ (IBM System 360 и др.). Таким образом, появление в кристалле i8080 механизма прямого доступа к памяти (ПДП) при работе с ВУ можно смело считать первым (но далеко не последним) ударом микропроцессоров по большим системам. ПДП открыл зеленую улицу для применения в микроЭВМ таких сложных устройств, как накопители на магнитных дисках и лентах, а также дисплеи на ЭЛТ, которые и превратили микроЭВМ в полноценную вычислительную систему.

Начиная с первого кристалла, Intel стала выпускать не отдельные чипы, а семейства БИС, рассчитанные на совместное использование. Помимо МП, в новый набор микросхем вошли ИС системных генератора и контроллера. Вскоре их пополнили БИС

контроллера ПДП и контроллера прерываний. Благодаря хорошо продуманному составу комплекта, проектирование МПС на его базе в ряде случаев упростилось.

Следует отметить, что в эти годы разработчики систем все большее внимание стали уделять развитию мультипроцессорным универсальным МПС, которые в ту пору еще не стали привычным атрибутом вычислительных центров. Поэтому в начале 1976 г. стартовали работы по созданию 16-разрядного прибора, который впоследствии получил обозначение i8086.

МП i8086. Конечной целью нового проекта было получение 16-разрядного микропроцессора с производительностью, на порядок превышающей аналогичный параметр кристалла i8080 и позволяющего создавать многопроцессорные системы. Поставленная задача решалась за счет дальнейшего совершенствования архитектурных концепций, положенных в основу его предшественника. Был разработан сложный и исключительно удачный процессор в очень сжатые сроки.

Новый кристалл был анонсирован 8 июня 1978 г. Прибор изготавливался по высококачественной трехмикронной МОП-технологии с кремниевыми затворами (Н-MOS), позволившей разместить на кристалле 29 тыс. транзисторов. Высокое быстродействие элементов (задержка 2 нс/вентиль) обеспечило тактовую частоту процессора 5 МГц, а 16-разрядная архитектура и 200-нс машинный цикл - производительность процессора, превышающую аналогичный параметр i8080 на порядок величины.

Программная совместимость с i8080 была, пожалуй, единственной, но вместе с тем и исключительно важной характеристикой, которая объединяла 86-й кристалл с его предшественниками. Структура процессора оказалась полностью пересмотренной. Прежде всего, прибор был разбит на два функциональных блока - операционный и интерфейсный, которые могли работать одновременно. В результате исполнение одной команды совмещалось во времени с выборкой следующей инструкции или данных из памяти. Более того, в МП появился регистровый файл команд, что давало дополнительную экономию времени при обращениях к памяти. Алгоритм работы операционного и интерфейсного блоков позволял вести обработку команд, находящихся в конвейере команд

регистрового файла, одновременно с их вводом из программной памяти.

Возможность адресации 1 Мбайт ОЗУ и сегментация памяти могут быть отнесены к одним из наиболее существенных новшеств, предложенных инженерами Intel. В частности, сегментация памяти и большое число уровней прерываний были ориентированы на работу систем в многозадачном режиме, весьма актуальном для приложений управления.

Большая емкость ОЗУ позволяла перевести проекты построения сложных операционных и прикладных систем из области теории в сферу практической реализации. Наряду с поддержкой ввода/вывода по каналу прямого доступа к памяти i8086 обеспечивал адресацию до 64К портов программно-управляемого ввода/вывода. Это снимало практически любые ограничения при формировании крупных систем сбора и обработки информации.

Микропроцессор имел два режима работы - минимальный и максимальный. Первый рассчитан на его использование в однопроцессорных системах и предполагал работу кристалла без БИС контроллера шины. Максимальный режим был ориентирован на применение МП в многопроцессорных системах и требовал наличия указанного контроллера. Таким образом, один и тот же процессор с одинаковым успехом мог применяться в системах различного класса.

Система команд процессора содержала 147 инструкций. Она позволяла решать задачи управления практически любой сложности. Появление среди них таких операций, как умножение и деление 16-разрядных чисел со знаком и без знака, команд обработки массивов данных, а также программно-управляемых прерываний дает все основания назвать этот кристалл универсальным, рассчитанным на использование не только в сложных контроллерах, но и в качестве центрального процессора ЭВМ общего назначения.

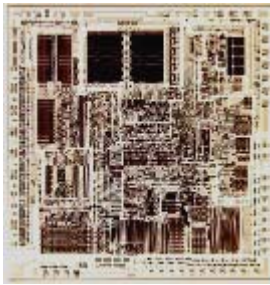
МП вышел в мощном сопровождении средств поддержки: вспомогательных БИС, средств разработки и отладки аппаратуры и системного ПО.

Использование микросхем i8086 в персональных компьютерах IBM предопределило дальнейшее развитие корпорации Intel

как разработчика и изготовителя универсальных процессоров общего назначения.

Вычислительная мощь 16-разрядных приборов была поддержана арифметическим сопроцессором i8087, который позволил превратить МПС в достаточно мощный инструмент и для решения задач вычислительного характера. Более того, теперь и разработчики систем управления на базе 86-го МП получили возможность использовать интенсивную арифметическую обработку информации, для которой ранее служили мини-ЭВМ.

3.2. Микропроцессоры второго поколения



МП i80286 второго поколения был разработан к 1 февраля 1982 г. Оставшись 16-разрядным прибором, по производительности новый ЦП в 3 - 6 раз превзошел своего предшественника при тактовой частоте первой модификации 8 МГц. Благодаря использованию многовыводного корпуса разработчики смогли применить схему с отдельными шинами адресов и данных.

24 разряда адреса позволили обращаться к физической памяти объемом до 16 МБайт. Встроенная система управления памятью и средства ее защиты открывали широкие возможности использования МП в многозадачных средах. Кроме того, аппаратура i80286 обеспечивала работу с виртуальной памятью объемом до 1 Гбайт. Для поддержки устройства управления памятью система команд пополнилась еще 16 инструкциями.

МП имел два режима работы - реальный и защищенный. В первом случае он воспринимался как быстрый МП i8086 с несколько расширенной системой команд и прекрасно подходил тем потребителям, для которых, помимо скоростных характеристик, жизненно важным было сохранение существующего задела ПО.

Работа в защищенном режиме позволяла использовать преимущества МП в полном объеме, и, прежде всего, большой объем основной памяти, дающий возможность работать ему в многозадачном варианте. Ведь основная проблема многозадачности была

в том, что предыдущие модели МП исполняемые программы могли быть записаны по любому адресу памяти, даже в занятые ячейки памяти ранее исполнявшимися программами. Операционная система и другие приложения при этом были не защищены: в любой момент исполняемая программа могла затереть эти места в памяти и система не смогла бы в дальнейшем вести достоверные расчеты.

Со стороны разработчиков программных продуктов были попытки создать операционную систему, которая сама бы контролировала все действия программ. Но для этого пришлось отказаться от компиляции приложений в готовые машинные коды - они стали интерпретируемыми, а производительность упала раз в двадцать. Стало ясно, что без аппаратной акселерации контроля, т. е. без защищенного режима процессора не обойтись.

Суть работы защищенного режима состоит в следующем. Все свои команды процессор выполнял точно так же как и в реальном режиме, но программистам пришлось использовать понятие «логического адреса». Логический адрес состоял из 32 бит: селектора (16 бит) и смещения (16 бит) (рис. 2). При этом в сегментных регистрах теперь хранился не сегмент, а селектор. Селектор - это индекс в таблице дескрипторов.

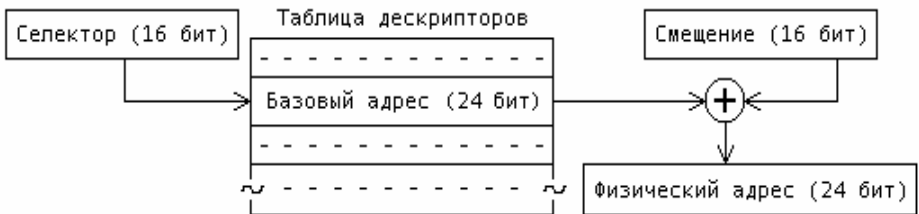


Рис. 2

Запись в таблице дескрипторов содержала всю необходимую информацию о некотором блоке памяти: его базовый адрес, размер всего блока, его тип (код или данные) и сведения о приоритете программы - владельца. Таким образом, каждый дескриптор полностью описывал один сегмент программы. Размер этого дескриптора был одинаков как для 286-х, так и для 386-х машин - 64 бита или 8 байт, но у 286-го старшие 16 бит не использовались.

Существовала одна глобальная и несколько локальных таблиц. Глобальная присутствовала всегда и хранила информацию о сегментах операционной системы. Локальные таблицы были для всех остальных программ. Управление памятью в защищенном режиме всегда было связано с конкретной операционной системой и ее версией. В операционной системе (ОС) OS/2 2.0 каждой программе были доступны глобальная и локальная (своя) таблицы дескрипторов. Всем приложениям в ОС Windows 3.0 давалась одна общая локальная таблица.

Всеми преимуществами МП решила воспользоваться IBM, применив процессор в новой модели ПК типа AT.

К сожалению, защищенный режим 286-го обладал и недостатками: несмотря на возможность адресовать 16 Мб памяти, максимальный размер сегмента остался по-прежнему равным 64 Кб, затрудняя программистам работу с большими массивами данных.

Режим работы с виртуальной памятью имел недостаток. Он заключался в том, что отсутствовал «прозрачный» для приложений способ перемещения данных операционной системой из памяти на жесткий диск - для реализации этого программам приходилось прибегать к разным ухищрениям вроде «запираания» и «отпираания» указателей на блок памяти.

В защищенном режиме отсутствовала совместимость с программами, написанными для реального режима MS-DOS. Переход из реального режима в защищенный был односторонним, для обратного перехода требовалась перезагрузка системы.

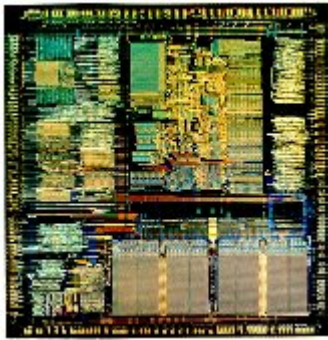
Указанные недостатки и высокая потребность в высокопроизводительных МП стимулировали усилия специалистов Intel по разработке прибора следующего поколения. Увеличение тактовой частоты 286-го процессора сверх достигнутого предела в 16 МГц давалось уже слишком дорого, а кроме того, никак не устраняло узкого места системы, которым оставалась оперативная память. Помимо прочего, 286-й решил далеко не все проблемы, характерные для многозадачных сред.

У инженеров Intel было два пути кардинального повышения производительности процессора: 32-разрядная архитектура прибора и совершенствование тракта процессор - память. При этом эффективное функционирование МП под управлением многоза-

дачных ОС требовало усовершенствования устройства управления памятью.

3.3. Микропроцессоры третьего поколения

МП i80386 третьего поколения был представлен 17 октября 1985 г. Использование КМОП-технологии с проектными нормами 1 мкм и двумя уровнями металлизации позволило разместить на кристалле 275 тыс. транзисторов и реализовать полностью 32-разрядную архитектуру МП.



32 разряда адреса обеспечили адресацию физической памяти объемом до 4 Гбайт и виртуальной памяти емкостью до 64 Тбайт. Встроенная в МП система управления памятью и защиты включала регистры преобразования адреса, механизмы защиты оперативной памяти и улучшенные аппаратные средства поддержки многозадачных ОС.

Помимо работы с виртуальной памятью допускались операции с памятью, имевшей страничную организацию. Предварительная выборка команд, буфер на 16 инструкций, конвейер команд и аппаратная реализация функций преобразования адреса значительно уменьшили среднее время выполнения команды.

Благодаря этим архитектурным особенностям, процессор мог выполнять 3 - 4 млн. команд в секунду, что примерно в 6 - 8 раз превышало аналогичный показатель для МП i8086. Безусловно, новый прибор остался совместимым со своими предшественниками на уровне объектных кодов.

Одной из наиболее любопытных особенностей рассматриваемой разработки компании было использование высокоскоростной кэш-памяти, позволившей существенно повысить производительность систем на базе 386-го процессора (еще один атрибут универсальных машин, который стал применяться в микропроцессорных системах). Для управления работой этой памятью была разработана БИС высокопроизводительного контроллера кэш-

памяти типа i82385, с помощью которой формировался двухходовой множественный ассоциативный кэш. Указанная БИС обеспечивала управление памятью емкостью до 32 Кбайт и высокий коэффициент удачных обращений.

Для реализации работы с числами с плавающей точкой был разработан математический сопроцессор, который выпускался в виде отдельного кристалла i80387, дополняя вычислительную мощь МП.

Особый интерес представляли три режима работы кристалла - реальный, защищенный и режим виртуального МП i8086. В первом обеспечивалась совместимость на уровне объектных кодов с устройствами i8086 и i80286, работающими в реальном режиме. При этом архитектура i80386 была почти идентична архитектуре 86-го процессора, для программиста же он вообще представлялся как МП i8086, выполняющий соответствующие программы с большей скоростью и обладающий расширенными системой команд и регистрами.

Одно из основных ограничений реального режима на практике было связано с предельным объемом адресуемой памяти, равным 1 Мбайт. От него свободен защищенный режим, позволяющий воспользоваться всеми преимуществами архитектуры нового МП. Размер адресного пространства в этом случае увеличивался до 4 Гбайт, а объем поддерживаемых программ - до 64 Тбайт.

Производителям ПО это позволяло задействовать достаточно гибкие методы разработки и создавать более крупные программные пакеты. Для конечных пользователей выполнение приложений, рассчитанных на работу в реальном и защищенном режимах, происходило без каких-либо функциональных отличий, поскольку управление обоими режимами базировалось на средствах ОС и специальном прикладном ПО. Однако системы защищенного режима обладали более высоким быстродействием и возможностями организации истинной многозадачности.

Наконец, режим виртуального МП открывал возможность одновременного исполнения ОС и прикладных программ, написанных для МП i8086, i80286 и i80386. Поскольку объем памяти, адресуемой 386-м процессором, не ограничен значением 1 Мбайт,

он позволял формировать несколько виртуальных сред i8086. Немаловажно, что эти среды могли порождаться в одно и то же время, а механизм защищенного режима обеспечивал ОС и ее прикладным задачам использование различных областей памяти. Благодаря таким возможностям аппаратуры, можно было вместо нескольких МП типа i8086 использовать один процессор i80386, сохранив львиную долю имевшегося ПО.

Примерно в этот же период IBM и Microsoft приступили к разработке новой многозадачной ОС с графическим интерфейсом пользователя.

Стремление удовлетворить запросы потребителей всех категорий привело Intel к созданию клона 386-го МП с 16-разрядной внешней шиной данных (при сохранении внутренней 32-разрядной архитектуры). Существующий прибор получил обозначение i80386SX и был анонсирован 16 июня 1988 г., а уже менее чем через полгода пользователям были предложены первые ПК на его основе. Поскольку эти модели стоили дешевле компьютеров с МП 80386DX, многие потребители вполне справедливо рассматривали их как начальную ступень в применении вычислительной техники.

В конце 80-х годов степень интеграции микросхем приближалась к 1 млн. транзисторов на кристалле и 10 апреля 1989 г. Intel объявила о начале выпуска 32-разрядного прибора четвертого поколения - i80486, ставшего после устройств i8080 и i8086 еще одним долгожителем.

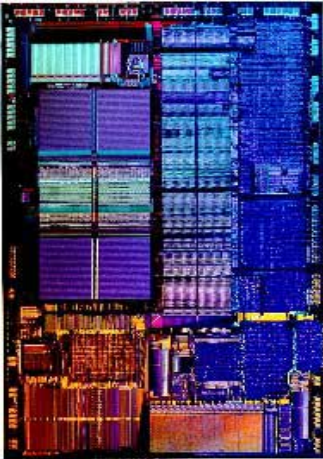
3.4. Микропроцессоры четвертого поколения

МП i80486. Архитектура нового МП отчасти напоминала строение своего предшественника, но вместе с тем имела и ряд коренных отличий. 1,2 млн. транзисторов позволили разработчикам реализовать на кристалле быстродействующую кэш-память (L1) и математический сопроцессор. Такое техническое решение свело к возможному минимуму число чипов на плате и самым благоприятным образом сказалось на стоимости готовых систем.

В 486-м процессоре кэш-память имела объем 8 Кб и была предназначена для одновременного хранения данных и инструкций. Кэш-память имела 4-канальную наборно-ассоциативную ар-

хитектуру и работала на уровне физических адресов памяти. Она содержала 128 наборов по 4 строки размером по 16 байт. Кэш-память умела работать только со строками, и если процессор требовал какой-нибудь байт, отсутствующий в кэше, то кэш-контроллер загружал из ОЗУ всю 16-байтную строку, содержащую необходимый байт.

Выбор строки для замещения производился по алгоритму «псевдо-LRU», для этого каждому набору строк отводилось по 3 бита статистики использования. Алгоритм LRU (Least Recently Used) основан на поиске элемента, к которому дольше всего не было обращений. При каждом обращении к строке кэш-контроллер увеличивал на 1 соответствующий счетчик LRU. Приставка «псевдо» означает лишь несовершенство механизма работы, ведь под счетчик отводилось всего 3 бита, что дает всего 8 состояний счетчика (2^3). После 8-го обращения к строке счетчик обнулится и соответствующая строка из самой «необходимой» станет самой «не необходимой» и будет прямым кандидатом на замещение.



Кэш-память первых 486-х работала в режиме Write Through (сквозная запись). В этом случае при записи данных тратилось дополнительное время на их запись во внешнюю память (даже если они присутствовали в кэше). Эта давала возможность ускорить чтение данных, но скорость записи, при этом, не ускоряется.

В следующих модификациях 486-х процессоров (некоторые 486DX2 и все 486DX4) был реализован принцип Write Back. В этом варианте запись данных, если их старая копия уже присутствовала в кэш-памяти, производилась только в кэш-память, а запись в ОЗУ откладывалась.

Процессор i486 мог использовать и внешнюю кэш-память (L2), расположенную вне кристалла микросхемы процессора. В 486-м, как видно, появилось 2-х уровневой кэш. Очевидно, что даже если оба кэш работают на одной частоте, кэш-память L1

функционирует быстрее второго. Это связано с тем, что при чтении данных из кэш-памяти L2 процессор все равно вынужден делать несколько пустых тактов, хотя и меньше, чем при чтении из ОЗУ.

Объем L2 составлял от 256 до 512 Кб. В системных платах 386-х моделей L2 обычно не превышал 128 Кб (типичный объем - 64 Кб). В марте 1994-го Intel, выпустив 486DX4, увеличила объем L1 до 16 Кб, при этом он по-прежнему оставался общим для данных и для команд.

МП функционировал в трех режимах и был ориентирован на многозадачные среды. За счет интеграции математического сопроцессора в БИС, а также модернизации его архитектуры производительность на задачах вычислительного характера возросла в 3 - 4 раза. Общая же производительность 486-го превышала аналогичный параметр своего предшественника в 4 - 5 раз.

Ровно через два года после выпуска i80486 появилась упрощенная версия кристалла (без сопроцессора), получившая обозначение i80486SX. Дальнейшее совершенствование пошло по пути увеличения тактовой частоты: были представлены версии на 50, 66, 75 и 100 МГц.

3.5. Микропроцессоры пятого поколения

Pentium P5. Выпуск высокопроизводительных МП Pentium P5 началось 22-го марта 1993 года. Это был первый процессор с двухконвейерной структурой. Он имел тактовые частоты 60 и 66 МГц. Частота шины совпадала с тактовой частотой процессора.



Процессоры содержали более 3.1 млн. транзисторов и выпускались по технологии 0.80 мкм, а позже – 0.60 мкм. Размер L1 составлял 16 Кб - 8 Кб на данные и 8 Кб на инструкции. L2 размещался на материнской плате и мог иметь объем до 1 Мб. Процессор выпускался для разъема Socket 4.

Как показали результаты эксплуатации МП, простое наращивание тактовой частоты неоднозначно влияло на увеличение его

производительности. В первую очередь это связано с большим разбросом во времени выполнения различных программ (приложений), связанных с разницей в производительности их различных компонентов. Так, например, разброс высоких и низких значений производительности относительно среднего значения компьютеров, построенных на процессорах 80486 и Pentium P5, составляет ориентировочно от 10 % до 20 % (рис. 3).

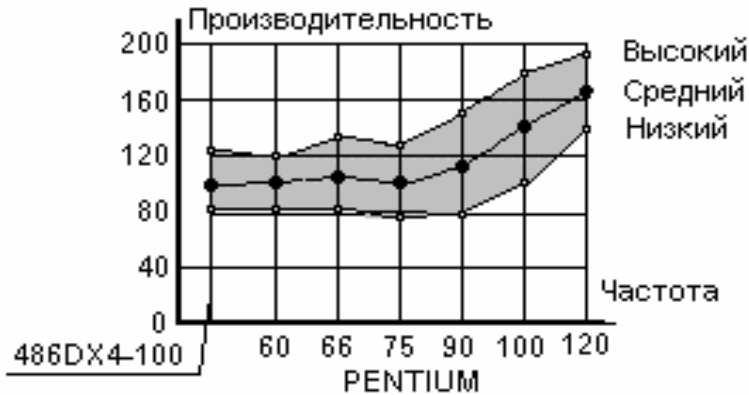


Рис. 3

Оценка производительности здесь проводилась по индексу iCOMP (Intel Comparative Microprocessor Performance), который учитывает четыре главных аспекта производительности процессора при 32- и 16 - разрядных операциях: с целыми числами, числами с плавающей точкой, графикой и видео.

Поэтому разработчики Intel основное внимание уделяли не только повышению тактовой частоты МП, но и совершенствованию его архитектуры. Эта тенденция сохранилась у разработчиков МП и в настоящее время.

В марте 1994 года Intel выпустила МП **Pentium P54**. Процессор имел частоты от 75 до 200 МГц. Частота шины 50-66 МГц. Размер L1 остался прежним – 16 Кб (8 Кб на данные и 8 Кб на инструкции). L2 был расположен вне кристалла и мог иметь объем до 1 Мб. При производстве этого процессора Intel применяет более совершенный техпроцесс 0.50 мкм. Процессор содержал более 3.3

млн. транзисторов. Выпускался для разъема Socket 5, позднее Socket 7.

Pentium MMX (P55) был выпущен 8 января 1997 года. Он пришел на смену МП P54 в связи с появлением все большего числа мультимедийных приложений. В нем был реализован новый набор из 57 команд MMX (Multi Media eXtention), существенно увеличивающий производительность компьютера при работе с этими приложениями (от 10 до 60 %, в зависимости от оптимизации).



МП выпускался с тактовыми частотами 166, 200 и 233 МГц. Тактовая частота шины составляла 66 МГц. По сравнению с Pentium P54 в нем был вдвое увеличен размер L1, который составил 32 Кб. Как и в предыдущих версиях L1 был разбит на два блока по 16 Кб для хранения данных и для инструкций.

L2 находился на материнской плате и мог иметь объем до 1 Мб. Процессоры выпускались по 0.35 мкм технологии и состояли из 4.5 млн. транзисторов. Он рассчитан на использование с разъемом Socket 7.

3.6. Микропроцессоры шестого поколения

Pentium PRO (P6) был выпущен 1 ноября 1995 года. От предыдущего поколения их отличало применение технологии динамического исполнения инструкций -



изменения порядка их исполнения, и архитектура двойной независимой шины. Добавилась еще одна шина, которая соединила процессор с L2, встроенным в ядро. В результате этого впервые был применен L2, работающий на частоте процессора. Первоначальный размер L2 имел 256 Кб, позже достиг 1024 Кб. Максимальный размер – 2048 Кб. L1 остался прежним: 8 Кб + 8 Кб.

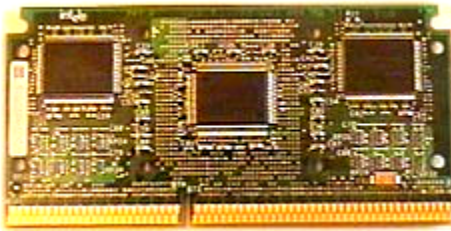
МП имел тактовые частоты 150, 166, 180, 200 МГц.

Процессоры Pentium PRO выпускались в корпусах SPGA (Staggered Pin Grid Array) с матрицей штырьковых выводов. В одном корпусе было установлено два кристалла – ядро процессора и L2 собственного изготовления. Устанавливался в Socket 8 с возможностью объединить до 4-х процессоров для симметричной мультипроцессорной обработки. Шина работала на частоте 60-66 МГц.

При 32-битных вычислениях и многозадачности значительно превосходил по производительности предыдущие версии Pentium, но в 16-битных приложениях проигрывал ему.

Процессор с тактовой частотой 150 МГц производился с использованием техпроцесса 0.60 мкм, более поздние модели – 0.35 мкм. Кристалл самого процессора состоял из более чем 5.5 млн. транзисторов, кэш-память содержала от 15.5 до 31 млн. транзисторов.

Pentium II (Klamath) появились 7 мая 1997 года. Эти процессоры объединили архитектуру Pentium PRO и технологию MMX. По сравнению с Pentium Pro удвоен размер L1 (16 Кб + 16 Кб). В процессоре была использована новая технология корпусов - картридж с печатным краевым разъемом, на который выведена



системная шина: SECC (Single Edge Contact Cartridge). Выпускался в конструктиве Slot 1. На картридже размером 14×6.2×1.6 см установлена микросхема ядра процессора, несколько микросхем, реализующих L2, и вспомогательные дискретные элементы (резисторы и конденсаторы).

Такой подход можно считать шагом назад – у Intel уже была отработана технология встраивания в ядро кэша второго уровня. Но таким образом можно было использовать микросхемы памяти сторонних производителей. В свое время, Intel считала такой подход перспективным на ближайшие 10 лет, хотя через непродолжительное время отказалась от него.

В то же время сохранилась независимость шины L2, которая тесно связана с ядром процессора собственной локальной шиной. Частота этой шины была вдвое меньше частоты ядра. Так что Pentium II имел большую L2, работающую на половинной частоте процессора.

Pentium II насчитывал около 7.5 млн. транзисторов только в процессорном ядре и выполнялся по технологии 0.35 мкм. Он имел тактовые частоты ядра 233, 266 и 300 МГц при частоте системной шины 66 МГц. При этом L2 работал на половинной частоте ядра и имел объем 512 Кб.

Для этих процессоров был разработан Slot 1, по составу сигналов схожий с Socket 8 для Pentium Pro. Однако Slot 1 позволяет объединять лишь пару процессоров для реализации симметричной мультипроцессорной системы, либо системы с избыточным контролем функциональности.

26 января 1998 году вышел процессор из линейки **Pentium II (Deschutes)**. От Klamath отличался более тонким технологическим процессом – 0.25 мкм и частотой шины 100 МГц. Имел тактовые частоты 350, 400, 450 МГц. Выпускался в конструктиве SECC, который в старших моделях был сменен на SECC2 - кэш с одной стороны от ядра, а не с двух, как в стандартном Deschutes и измененное крепление кулера.

Процессор состоял из 7.5 млн. транзисторов и выпускался для разъема Slot 1.

Pentium II OverDrive – процессор, вышедший 11 августа 1998 года, был предназначен для замены Pentium PRO на старых материнских платах. Он носил кодовое имя **P6T**. Имел частоту 333 МГц. Кэш первого уровня – 16 Кб на данные + 16 Кб на инструкции, кэш второго уровня имел размер 512 Кб, был интегрирован в ядро и работал на частоте процессора. Шина тактировалась частотой 66 МГц.



МП содержал 7.5 млн. транзисторов и производился по техпроцессу 0.25 мкм.

Микропроцессоры Celeron (Covington) стали новой веткой в направлении технологии микропроцессоров, направленной

на удешевление своей продукции. Он был выпущен как альтернативный вариант Pentium II, имевший довольно высокое соотношение «цена-производительность». МП был выпущен 15 апреля 1998 года и работал на тактовой частоте 266 МГц.

Этот процессор по числу устройств в нем был «усеченным» Pentium II. Celeron был построен на базе ядра Deschutes и не имел кэш-памяти второго уровня. Это привело к снижению его производительности, но и существенно снизило его стоимость. Celeron работал на шине 66 МГц и повторял все основные характеристики своего предка – Pentium II Deschutes: L1 – 16 Кб + 16 Кб, MMX, техпроцесс 0.25 мкм. 7.5 млн. транзисторов. Процессор выпускался без защитного картриджа - конструктив – SEPP (Single Edge Pin Package). Разъем - Slot 1.

Начиная с частоты 300 МГц, появились процессоры **Celeron (Mendocino)** с интегрированным в ядро L2, работающим на частоте процессора, размером 128 Кб. Он вышел 8 августа 1998.



Благодаря высокоскоростному L1 имел хорошую производительность, сравнимую с Pentium II (при условии одинаковой частоты системной шины). Выпускались с тактовыми частотами от 300 до 533 МГц. 30 ноября 1998 года.

До 433 МГц выпускался в двух конструктивах: SEPP и PPGA. Некоторое время параллельно существовали Slot-1 (266 - 433 МГц) и Socket-370 (300А - 533 МГц) варианты, в конце концов, первый был вытеснен последним.

Celeron (Mendocino) был шагом к Pentium III, но, работая на шине 66 МГц, не мог показать все преимущества интегрированного высокоскоростного L1. Так как L1 был интегрирован в ядро, значительно увеличилось количество транзисторов, из которых состоит процессор - 19 млн. Техпроцесс остался прежним – 0.25 мкм.

Для мощных систем Intel выпустил 29 июня 1998 года МП **Pentium II Xeon** - серверный вариант процессора Pentium II, пришедший на смену Pentium PRO. Он производился на ядре Deschutes и отличался от Pentium II более быстрой и более емкой (есть варианты с 1 или 2 Мб) кэш-памятью второго уровня. Вы-

пускался в конструктиве SECC для Slot 2. Это тоже краевой разъем, но с 330 контактами, регулятором напряжения VRM, запоминающим устройством EEPROM. Способен работать в мультипроцессорных конфигурациях.

L2, как и в Pentium PRO, полноскоростной. Только здесь он находится на одной плате с процессором, а не интегрирован в ядро. L1 – 16 Кб + 16 Кб. Частота шины – 100 МГц. Поддерживал набор инструкций MMX.

Процессор работал на частотах 400 и 450 МГц. Выпускался с применением техпроцесса 0.25 мкм. и содержал 7.5 млн. транзисторов.

Pentium III (Katmai) был разработан к 26 февраля 1999 года и мало чем отличался от Pentium II. Он работал на такой же шине с первоначальной частотой 100 МГц, позже появились модели, работающие на шине 133 МГц. МП выпускался в конструктиве SECC 2 и был рассчитан на установку в разъем Slot 1.

Кэш-память осталась прежней: L1 – 16 Кб + 16 Кб, а L2 – 512 Кб. Они были размещены на процессорной плате и работали на половинной частоте процессора.



Главным отличием МП является расширение набора SIMD-инструкций - SSE (Streaming SIMD Extensions). Также расширен набор команд MMX и усовершенствован

механизм потокового доступа к памяти.

Процессор работал на частотах 450-600 МГц, содержал 9.5 млн. транзисторов. Выпускался с применением техпроцесса 0.25 мкм.

Pentium III (Coppermine) был выпущен 25 октября 1999 года. По сути, именно Coppermine является новым процессором, а не доработкой Deschutes. Новый процессор имел полноскоростной интегрированный в ядро L2 размером 256 Кб (Advanced Transfer Cache).

Выпускался с использованием техпроцесса 0.18 мкм. Изменение технологии с 0.25 до 0.18 мкм позволило разместить на ядре большее число транзисторов и теперь их стало 28 млн. Правда, ос-

новная масса нововведенных транзисторов относится к интегрированному L2. Заметим, что L1 кэш остался без изменений.

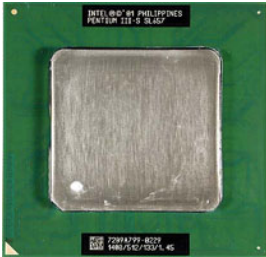


МП поддерживал наборы команд MMX и SSE. Сначала выпускался в конструктиве SECC 2, но так как кэш был встроен в ядро процессора, процессорная плата оказалась ненужной, и

только повышала стоимость процессора. Поэтому вскоре процессоры стали выходить в конструктиве FC-PGA (Flip-Chip PGA). Как и Celeron Mendocino, они работали в разьеме Socket 370.

Coppermine был последним процессором для Slot 1. Работал с шиной, имевшей частоту тактирования 100 и 133 МГц.

Pentium III (Tualatin) пришел на смену Coppermine 21 июня 2001 года. В это время на рынке уже присутствовали первые процессоры Pentium 4, и новый процессор был предназначен для испытания новой 0.13 мкм. технологии, а также для того чтобы заполнить нишу высокопроизводительных процессоров, так как производительность первых Pentium 4 была довольно низкой.

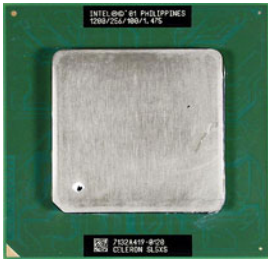


Tualatin - это изначальное название глобального проекта Intel по переводу производства процессоров на 0.13-микронную технологию. Сами процессоры с новым ядром стали первыми продуктами, появившимися в рамках этого проекта.

Изменений в самом ядре немного - добавилась только технология "Data Prefetch Logic". Она повышала производительность, предварительно загружая данные, необходимые приложению в кэш. Разъем для нового процессора остался прежним - Socket 370, а вот конструктив сменился на FC-PGA 2, который использовался в процессорах Pentium 4. От старого FC-PGA он в первую очередь отличается тем, что ядро было покрыто теплоотводящей пластиной, которая также защищает его от повреждения при установке радиатора.

МП Tualatin работали на шине с частотой 133 МГц и состояли из 44 млн. транзисторов. Поддерживали наборы инструкций MMX и SSE. Процессор работал на частотах от 1 ГГц до 1.33 ГГц (Desktop Tualatin), и от 1.13 ГГц до 1.4 ГГц (серверный вариант).

Celeron (Coppermine Lite) были разработаны 29 марта 2000 года с целью, чтобы не терять позиций на рынке бюджетных процессоров. Теперь это были абсолютно другие процессоры –



Intel повторил опыт создания первых процессоров с названием Celeron: использовал ядро процессора Pentium III с обрезанным до 128 кб L2 и медленной шиной 66 МГц.

Как видно из названия, процессор выполнен на ядре Coppermine с вдвое уменьшенным L2. Как и Pentium III Coppermine, новый Celeron, имел набор дополнительных команд SSE, быструю встроенную L1 и производится по той же технологической норме (0.18 мкм.), отличаясь только объемом L2 - 128 Кб против 256 Кб у Pentium III. Работает в том же разъеме Socket 370.

Первые процессоры появились с частотой 566 МГц и работали на шине 66 МГц. Позже, 3 января 2001 года, с выходом 800 МГц версии, Celeron перешел на более быструю 100 МГц шину. Максимальная частота этих процессоров составляла 1100 МГц. Кэш первого уровня: 32 Кб (16 Кб на данные и 16 Кб на инструкции). Процессор состоял из 28.1 млн. транзисторов.



2-го октября 2001 года, Intel переводит процессор **Celeron** на новое ядро – **Tualatin**. Еще никогда Celeron не был так близок к процессору Pentium. От Pentium III он отличался лишь более медленной 100 МГц шиной. В общем, оставив неизменным объем L2 и снизив частоту FSB до 100 МГц. Процессоры выпускались с тактовыми

частотами от 900 МГц до 1400 МГц, состояли из 44 млн. транзисторов, поддерживали MMX, SSE. Техпроцесс 0.13 мкм. Выпускались в конструктиве FC-PGA 2, для разъема Socket 370.

Pentium III (Tanner) был построен на базе Pentium III Katmai. Содержал 512, 1024 или 2048 Кб полноскоростной кэш памяти второго уровня. L1 - 16 Кб + 16 Кб. Выпускался с частотами 500 и 550 МГц с применением 0.25 мкм. техпроцесса и состоял из 9.5 млн. транзисторов. Работал на 100 МГц системной шине. Выпускался в конструктиве SECC для Slot 2. Был предназначен для использования в двух-, четырех-, восьмипроцессорных (и более) серверах и рабочих станциях.

С переходом Pentium III на новое ядро 25 октября 1999 года появилась и модификация МП **Xeon (Cascades)**. По сути, это было модернизированное ядро Coppermine. Процессор имел от 256 КБ до 2048 Кб кэш памяти второго уровня, работал на частотах системной шины 100 и 133 МГц (в зависимости от версии). Выпускались процессоры с частотами от 600 до 900 МГц. Процессоры с частотой 900 МГц из первых партий перегревались и их поставки были временно приостановлены. Как и предшественник, Xeon Cascades был рассчитан на установку в разъем Slot 2. Выпускался с применением 0.18 мкм. техпроцесса и состоял из 28.1 млн. транзисторов

Pentium 4 с NetBurst Micro-Architecture были предназна-



чены для работы на частотах порядка нескольких гигагерц, Intel увеличило длину конвейера Pentium 4 до 20 ступеней (Hyper Pipelined Technology) за счет чего удалось даже при технологических нормах 0,18 мкм добиться работы процессора на частоте в 2 ГГц. Однако из-за такого увеличения длины конвейера время выполнения одной

команды в процессорных тактах также сильно увеличивается. Поэтому компания провела доработку алгоритмов предсказания переходов (Advanced Dynamic Execution).

L1 в процессоре претерпела значительные изменения. В отличие от Pentium III, она могла хранить и команды, и данные.

Pentium 4 имел всего 8 Кб кэш данных. Команды, сохраняются в так называемом Trace Cache. Там они хранятся уже в декодированном виде, т.е. в виде последовательности микроопераций, поступающих для выполнения в исполнительные устройства процессора. Емкость этого кэша составляет 12000 микроопераций.

В новом процессоре был расширен набор команд - SSE2. К 70 инструкциям SSE, добавились еще 144 новые инструкции. Одной из множества инноваций была совершенно новая 100 МГц шина, передающая по 4 пакета данных за такт - QPB (Quad Pumped Bus), что дает результирующую частоту 400 МГц.

Первым из линейки Pentium 4 был МП **Willamette 423**. Появившись 20 ноября 2000 года с частотами 1.4 и 1.5 ГГц, эти процессоры, изготовленные с применением техпроцесса 0.18 мкм, достигли частоты 2 ГГц.

Процессор устанавливался в новый разъем Socket 423 и



выпускался в конструктиве FC-PGA 2. Он состоял из 42 млн. транзисторов.

Кэш 2-го уровня остался прежнего объема - 256 Кб. Ширина шины L2 составляет 256 бит, но латентность

кэш-памяти уменьшилась в два раза, что позволило добиться пропускной способности кэша в 48 Гб при частоте 1.5 ГГц.

Так как архитектура нового процессора была ориентирована в первую очередь на рост частоты, то первые процессоры Pentium 4 показали крайне низкую производительность. В большинстве задач 1.4 ГГц процессор уступал Pentium III Coppermine, работающему на частоте 1000 МГц.

27 августа 2001 года, появились МП **Willamette** предназначенные для установки в новый разъем - Socket 478. Процессор повторял все характеристики своего предка, за исключением конструктива - mPGA и разъема Socket 478.

Размеры процессора уменьшились благодаря тому, что теперь выводы сделаны непосредственно под ядром процессора.

Этот процессор, как и предшественник, работал на частотах от 1.4 до 2.0 ГГц.

Pentium 4 Northwood – так называется следующее ядро, на котором выпускались процессоры Pentium 4. Переход на 0.13 мкм. техпроцесс позволил еще больше наращивать тактовую частоту, и увеличить кэш второго уровня до 512 Кб. Увеличилось и количество транзисторов, которые составляют процессор – теперь их стало 55 млн. Естественно, что осталась поддержка наборов инструкций MMX, SSE и SSE2.

Первые процессоры на ядре Northwood появились 7 августа 2001 года с частотой 2.0 ГГц и частотой системной шины 400 МГц (4×100 МГц). МП Northwood, работают на частотах от 1.6 до 3.2 ГГц.

6-го мая 2002 года, Intel выпустила процессор на базе ядра Northwood с частотой системной шины 533 МГц (4×133 МГц) и тактовой частотой 2.26 ГГц. Так как модели с частотой шины 400 МГц выпускались с частотами до 2.6 ГГц, то и тут была применена буквенная маркировка.

14 апреля 2003 года выпускается процессор на все том же ядре Northwood, но уже с частотой системной шины 800 МГц (4×200 МГц) и тактовой частотой 3.0 ГГц. Позже, процессоры с 800 МГц системной шиной стали выпускаться с меньшими частотами – от 2.4 ГГц.

Pentium 4 XEON были представлены Intel 21 мая 2001 года, который базировался на ядре Willamette. Процессор выпущен в трех вариантах: 1.4 ГГц, 1.5 ГГц и 1.7 ГГц. Ядро процессора почти полностью идентично обычной версии Pentium 4 за исключением незначительных деталей. Это означает, что новый Хеон имеет все то, что есть в Pentium 4 – как достоинства новой архитектуры, так и ее недостатки.

Первые модели Хеон выпускались с применением 0.18 мкм. техпроцесса, с ядром, практически полностью повторявшим Pentium 4 Willamette и носившем кодовое имя **Foster**. Процессор выпускался с тактовыми частотами до 2,0 ГГц. Он состоял из 42 млн. транзисторов.



Кэш памяти первого уровня, как и у всех процессоров линейки Pentium 4, с архитектурой NetBurst, 8 Кб кэш данных. Кэш второго уровня – 256 Кб с улучшенной передачей данных (256 Кб Advanced Transfer Cache). Также как в Pentium 4 Willamette, в новом Хеон применена 400 МГц системная шина (4×100 МГц) которая синхронно работает с двумя каналами памяти на частоте 400 МГц.

Исторически, линейки процессоров Intel Хеон (то есть Pentium II Хеон, Pentium III Хеон) всегда использовали отличный от обычных версий процессора конструктив. В то время как процессоры Pentium II и Pentium III выпускались в 242-контактном Slot 1 варианте, то их Хеон версии использовали 330-контактный разъем Slot-2. Большинство добавочных ножек использовалось для снабжения кристалла дополнительной энергией.

С двумя мегабайтами L2 Pentium III Хеон потреблял больше энергии, чем его 256-килобайтный собрат. Аналогичная ситуация произошла и с новым Хеон. Если первые процессоры Pentium 4 Willamette, используют 423-контактный разъем, то в Хеон применяется 603-контактный интерфейс, предназначенный для использования в разъеме Socket 603.

Процессор мог работать только в одно- или двухпроцессорных конфигурациях.

9 января 2002 года появляются процессоры Хеон, сделанные на базе ядра Northwood с применением 0.13 мкм. техпроцесса, и оснащенные 512 Кб кэш памяти второго уровня. Кодовое название ядра – **Prestonia**.

От своего предшественника Хеон Foster отличается только увеличенной кэш-памятью и более совершенным техпроцессом.

Процессоры работали на частотах от 1.8 ГГц, до 3.0 ГГц и состояли из 55 млн. транзисторов. В процессорах с ядром Prestonia впервые появилась поддержка Hyper-Threading.

12 марта 2002 года, выходит процессор **Xeon MP**. Изготовлен с применением 0.18 мкм. и оснащен 256 Кб кэш памяти второго уровня. Основное отличие от процессоров Xeon Foster - возможность работать в многопроцессорных системах. Они работали на частотах от 1.4 до 1.6 ГГц. В этих процессорах осуществлена поддержка технологии Hyper-Threading (HT).

Сущность этой технологии заключается в том, что один физический процессор с Hyper-Threading видится системой как два, что позволяет оптимизировать загрузку его ресурсов и повысить производительность. В каждый момент времени только часть ресурсов процессора используется при выполнении программного кода. Неиспользуемые ресурсы также можно загрузить работой - например, задействовать для параллельного выполнения еще одного приложения (либо другого потока этого же приложения).

HT – это не настоящая многопроцессорность, ведь количество блоков непосредственно исполняющих команды не изменилось. Повысился лишь коэффициент их использования. Поэтому, чем лучше оптимизирована конкретная программа под HT, тем выше будет выигрыш в производительности.

По данным Intel, преимущество от HT может достигать 30%, в то время как блоки, ее реализующие, занимают менее 5% общей площади кристалла Pentium 4. Впрочем, даже идеально оптимизированные приложения могут, к примеру, обращаться к данным, которых нет в кэш-памяти процессора, заставляя его простаивать. Если сама архитектура NetBurst была рассчитана на повышение количества мегагерц, то Hyper-Threading наоборот, рассчитан на повышение выполняемой работы за один такт.

Одной из причин достаточно позднего представления Hyper-Threading в Pentium 4 (поддержка существует не только в ядре Northwood, но даже в Willamette, однако была заблокирована) являлась относительно небольшая распространенность Windows XP – единственной ОС семейства Windows, полноценно поддерживающей новую технологию. Также технологию должен поддерживать чипсет и BIOS системной платы.

Технологию Hyper-Threading поддерживает процессор Pentium 4 3.06 ГГц с частотой системной шины 533 МГц, а также все процессоры с частотой шины 800 МГц.

4 ноября 2002 года появляются процессоры Xeon MP, изготовленные с применением 0.13 мкм. техпроцесса. Эти процессоры, работающие на частотах 1.5 ГГц, 1.9 ГГц и 2.0 ГГц отличаются от своего собрата Xeon Prestonia не только возможностью работы в многопроцессорных конфигурациях, но и наличием интегрированной L3 размером 1 или 2 Мб. Благодаря этому увеличилось количество транзисторов, составляющих процессор до 108 млн.

18 ноября 2002 года появились процессоры Xeon работающие на 533 МГц (4 × 133 МГц) системной шине. Эти процессоры сделаны на ядре Prestonia, с применением 0.13 мкм. техпроцесса и состоят из 108 млн. транзисторов. Кэш - память второго уровня – 512 Кб, третьего уровня - 1 Мб. Процессоры Xeon на 533 МГц шине выпускаются с тактовыми частотами от 2.0 ГГц до 3.06 ГГц (вышел 10 марта 2003).

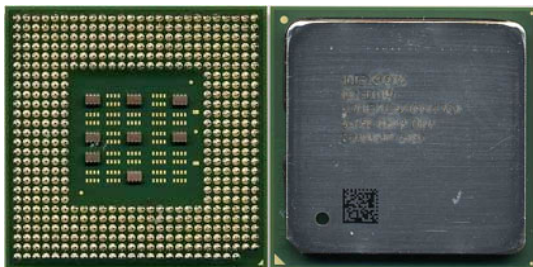
Pentium 4 HT начали выпускаться 14 ноября 2002 года. Он имеет частоту 3.06 ГГц, а системная шина тактировалась частотой 533 МГц с поддержкой технологии Hyper-Threading.

Celeron (Willamette 128) был выпущен с целью вытеснения с рынка процессоров для Socket 370, а также, желая занять нишу бюджетных процессоров (где до этого был Celeron Tualatin).

Ядро Willamette 128 архитектурно ничем не отличается от ядра Pentium 4 Willamette. Организация кэш - памяти и алгоритмы его работы не изменились, единственное отличие заключается в размере - 128 Кб кэш - памяти второго уровня вместо 256 Кб в оригинальном Pentium 4 Willamette.

Естественно, сохранен и форм-фактор Socket 478. 15 мая 2002 года появляется первый процессор с названием Celeron, построенный на базе Pentium 4, с частотой 1.7 ГГц. Позже, 12 июня 2002 года появляется версия на 1.8 ГГц.

Новый Celeron, как и раньше, использует 100 МГц системную шину, правда теперь уже с передачей 4-х сигналов за такт. Учетверенная 100 МГц системная шина наконец-то решает старую проблему Celeron - недостаток пропускной способности FSB.



Celeron выполнен с применением 0.18 мкм. техпроцесса. Состоит из 42 млн. транзисторов. Выпускается с частотами 1.7 и 1.8 ГГц.

Следующее ядро процессора Celeron, это **Northwood** (с урезанной до 128 Кб кэш - памятью второго уровня). Первым процессором на этом ядре был Celeron 2.0 ГГц, который вышел 18 сентября 2002 года. Он, как и Celeron Willamette 128, полностью повторяет характеристики старшего брата Pentium 4 Northwood, за исключением шины, рассчитанной исключительно на 400 МГц (4 × 100 МГц) и кэш - памятью второго уровня размером 128 Кб.

Применение 0.13 мкм. техпроцесса дает преимущество в виде хорошей разгоняемости. У ядра Northwood хороший частотный потенциал, поэтому запас для разгона есть.

В конце 2003 года Intel представила новое ядро для своих процессоров – **Prescott**. Эти процессоры изготовлены с применением 0.09 мкм. (90 нм) технологии. Ядро Prescott состоит из 125 млн. транзисторов, содержит 1 Мб кэш- - память второго уровня, увеличена кэш - память первого уровня до 32 Кб. Ядро обладает поддержкой технологии Hyper-Threading 2, дальнейшее развитие «многопроцессорности» в одном чипе.

В МП добавлен новый набор инструкций (или расширен уже присутствующий), включающий 15 новых инструкций по переводу чисел с плавающей запятой в целые, арифметику комплексных чисел, специальные команды для декодирования видео, SIMD-инструкции для формата с плавающей запятой и процесс синхронизации потоков.

Первые процессоры с этим ядром предназначены для работы на частотах 3.2 и 3.4 ГГц. Их корпуса совместимы с корпусами процессоров Pentium 4 Northwood.

На базе нового ядра продолжен выпуск процессоров линейки Celeron. Чипы **Celeron** на ядре **Prescott** быстрее предшественников на Northwood не только за счет возросшей тактовой частоты ядра. Они поддерживают системную шину с частотой 533 МГц, а объем их кэш - памяти увеличен со 128 до 256 кб. Celeron на ядре Prescott имеют частоты 2.8 и 3.06 ГГц.

Pentium 4 Extreme Edition оснащен технологией Hyper-Threading, работает на системной шине 800 МГц, имеет тактовую частоту ядра 3.2 ГГц. Но главным его отличием от предшествующих Pentium 4 стало наличие интегрированной в кристалл кэш-памяти третьего уровня L3 объемом 2 Мб. Эта кэш-память дополняет стандартный кэш L2 512 кбайт и работает также на частоте ядра процессора (правда, с гораздо большей латентностью, поскольку она асинхронная и призвана ускорять работу с данными из наиболее часто используемых областей системной памяти). Таким образом, Pentium 4 Extreme Edition имеет кэш-память объемом 2.5 Мб. А также является единственным desktop процессором с кэшем третьего уровня, интегрированным в ядро.

Процессор Pentium 4 Extreme Edition позиционируется Intel главным образом для игрового рынка, хотя не исключено и его применение в производительных рабочих станциях. Процессор использует ядро от мультипроцессорных Xeon MP с интегрированной кэш-памятью L3. Его немного изменили с целью поддержки системной шины 800 МГц, уменьшения энергопотребления и др. и упаковали в стандартный корпус от Pentium 4.

В настоящее время Intel ведет активные работы по созданию следующих поколений кристаллов с проектными технологическими нормами 65 нм. Также ведутся разработки и есть работающие чипы, изготовленные с применением не только 0.065 мкм. техпроцесса, но и 45 нм, 32 нм и даже 22 нм.

За Prescott планируется выпуск МП на ядре **Tejas** с шиной 1066 МГц. На его основе будут представлены восемь различных процессоров с тактовыми частотами от 6 до 9.2 ГГц. После этого компания планирует представить ядро **Nehalem**, использующее

системную шину 1200 МГц и позволяющее получить рабочую частоту свыше 10 ГГц. Nehalem будет основан на совершенно новой архитектуре. Это будет не модернизированный Pentium 4, как Prescott и Tejas. В нем будет применена система аппаратной защиты LaGrande, и по некоторым данным, использована более совершенная технология многопоточной обработки. Число транзисторов в чипе составит порядка 150-250 миллионов

4. Классификация микропроцессоров

Микропроцессор как функциональное устройство ЭВМ обеспечивает эффективное автоматическое выполнение операций обработки цифровой информации в соответствии с заданным алгоритмом. Для решения широкого круга задач в различных областях применений микропроцессор должен обладать алгоритмически полной системой команд (операций).

Теоретически показано, что минимальная алгоритмически полная система команд процессора состоит из одной или нескольких универсальных команд. Однако использование процессоров с минимальными по числу операций системами команд ведет к неэкономичному использованию информационных емкостей памяти и значительным затратам времени на выполнение «длинных» программ. Поэтому обычно в МП встраиваются аппаратные средства, позволяющие реализовать многие десятки и сотни команд. Такие развитые системы команд дают возможность обеспечить компактную запись алгоритмов и соответственно эффективные программы.

При проектировании МП решаются задачи определения наборов команд, выполняемых программным или аппаратным способом на основе заданной системы микрокоманд. Аппаратурная реализация сложных команд дает возможность увеличить быстродействие микропроцессора, но требует значительных аппаратных ресурсов кристалла интегральной схемы МП. Программная реализация сложных команд позволяет обеспечивать программирование сложных задач, изменять количество и особенности исполнения сложных команд. Однако скорость исполнения про-

граммных команд ниже скорости исполнения аппаратурно-реализованных команд.

Практически во всех современных МПС используются сложные развитые системы команд. Их ядро, состоящее из набора универсальных команд, реализуется аппаратурным способом в центральном МП. Кроме того, специализированные части наборов системы команд реализуются вспомогательными или периферийными микропроцессорами. Эти расширяющие возможности обработки данных специальные арифметические или логические МП позволяют ускорить выполнение определенных команд и тем самым сократить время исполнения программ.

Для описания МП как функциональных устройств необходимо охарактеризовать формат обрабатываемых данных и команд, количество, тип и гибкость команд, методы адресации данных, число внутренних регистров общего назначения и регистров результата, возможности организации и адресации стека, параметры виртуальной памяти и информационную емкость прямо адресуемой памяти. Большое значение имеют средства построения системы прерываний программ, построения эффективных систем ввода — вывода данных и развитого интерфейса.

МП могут быть реализованы на различной физической основе: на электронной, оптоэлектронной, оптической, биологической и даже на пневматической или гидравлической.

По назначению различают универсальные и специализированные микропроцессоры.

Универсальные МП предназначены для решения широкого круга задач. При этом их эффективная производительность слабо зависит от проблемной специфики решаемых задач. В системе команд МП заложена алгоритмическая универсальность, означающая, что выполняемый машиной состав команд позволяет получить преобразование информации в соответствии с любым заданным алгоритмом.

К универсальным МП относятся и секционные микропроцессоры, поскольку для них система команд может быть оптимизирована в каждом частном проекте создания секционного микропроцессора.

Эта группа МП наиболее многочисленна, в нее входят такие комплекты как K580, Z80, Intel 80×86, K582, K587, K1804, K1810 и др.

Специализированные МП предназначены для решения определенного класса задач, а иногда только для решения одной конкретной задачи. Их существенными особенностями являются простота управления, компактность аппаратных средств, низкая стоимость и малая мощность потребления.

Специализированные МП имеют ориентацию на ускоренное выполнение определенных функций, что позволяет резко увеличить эффективную производительность при решении только определенных задач.

Среди специализированных микропроцессоров можно выделить различные микроконтроллеры, ориентированные на выполнение сложных последовательностей логических операций; математические МП, предназначенные для повышения производительности при выполнении арифметических операций за счет, например матричных методов их выполнения; МП для обработки данных в различных областях применений и т. д.

С помощью специализированных МП можно эффективно решать новые сложные задачи параллельной обработки данных. Например, они позволяют осуществить более сложную математическую обработку сигналов, чем широко используемые методы корреляции, дают возможность в реальном масштабе времени находить соответствие для сигналов изменяющейся формы путем сравнения их с различными эталонными сигналами для эффективного выделения полезного сигнала на фоне шума и т. д.

По виду обрабатываемых входных сигналов различают цифровые и аналоговые микропроцессоры.

Сами МП являются цифровыми устройствами обработки информации. Однако в ряде случаев они могут иметь встроенные аналого-цифровые и цифро-аналоговые преобразователи. Поэтому входные аналоговые сигналы передаются в МП через преобразователь в цифровой форме, обрабатываются и после обратного преобразования в аналоговую форму поступают на выход.

С архитектурной точки зрения такие микропроцессоры представляют собой аналоговые функциональные преобразователи

сигналов. Они выполняют функции любой аналоговой схемы (например, производят генерацию колебаний, модуляцию, смещение, фильтрацию, кодирование и декодирование сигналов в реальном масштабе времени и т. д., заменяя сложные схемы, состоящие из операционных усилителей, катушек индуктивности, конденсаторов и т.д.). При этом применение аналогового МП значительно повышает точность обработки аналоговых сигналов и их воспроизводимость, а также расширяет функциональные возможности за счет программной “настройки” цифровой части микропроцессора на различные алгоритмы обработки сигналов.

Обычно в составе однокристалльных аналоговых МП имеется несколько каналов аналого-цифрового и цифро-аналогового преобразования. В аналоговом микропроцессоре разрядность обрабатываемых данных достигает 24 бит и более. Большое значение уделяется увеличению скорости выполнения арифметических операций.

Отличительная черта аналоговых МП - это способность к переработке большого объема числовых данных, т. е. к выполнению операций сложения и умножения с большой скоростью, при необходимости даже за счет отказа от операций прерываний и переходов. Аналоговый сигнал, преобразованный в цифровую форму, обрабатывается в реальном масштабе времени и передается на выход обычно в аналоговой форме через цифро-аналоговый преобразователь. При этом согласно теореме Котельникова частота квантования аналогового сигнала должна вдвое превышать верхнюю частоту сигнала.

Одним из направлений дальнейшего совершенствования аналоговых МП является повышение их универсальности и гибкости. Поэтому вместе с повышением скорости обработки большого объема цифровых данных будут развиваться средства обеспечения развитых вычислительных процессов обработки цифровой информации за счет реализации аппаратных блоков прерывания программ и программных переходов.

По количеству выполняемых программ различают одно- и многопрограммные микропроцессоры.

В однопрограммных МП выполняется только одна программа. Переход к выполнению другой программы происходит после завершения текущей программы.

В много- или мультипрограммных МП одновременно выполняется несколько (обычно несколько десятков) программ. Организация мультипрограммной работы микропроцессорных управляющих систем, например, позволяет осуществить контроль за состоянием и управлением большим числом источников или приемников информации.

По числу БИС в микропроцессорном комплекте различают однокристалльные, многокристалльные и многокристалльные секционные МП.

Процессоры даже самых простых ЭВМ имеют сложную функциональную структуру, содержат большое количество электронных элементов и множество разветвленных связей. Реализовать принципиальную схему обычного процессора в виде одной или нескольких БИС практически невозможно из-за специфических особенностей БИС (ограниченность количества элементов, сложность выполнения разветвленных связей, сравнительно небольшое число выводов корпуса). Поэтому необходимо изменять структуру процессора так, чтобы полная принципиальная схема или ее части имели количество элементов и связей, совместимое с возможностями БИС. При этом МП приобретают внутреннюю магистральную структуру, т. е. в них к единой внутренней информационной магистрали подключаются все основные функциональные блоки (арифметико-логический, рабочих регистров, стека, прерываний, интерфейса, управления и синхронизации и др.).

Для обоснования классификации МП по числу БИС надо распределить все аппаратурные блоки процессора между основными тремя функциональными частями: операционной, управляющей и интерфейсной. Сложность операционной и управляющей частей процессора определяется их разрядностью, системой команд и требованиями к системе прерываний; сложность интерфейсной части - разрядностью и возможностями подключения других устройств ЭВМ (памяти, внешних устройств, датчиков и исполнительных механизмов и др.). Интерфейс процессора содер-

жит несколько десятков шин информационных магистралей данных, адресов и управления.

Однокристалльные МП получаются при реализации всех аппаратурных средств процессора в виде одной БИС или СБИС. По мере увеличения степени интеграции элементов в кристалле и числа выводов корпуса параметры однокристалльных МП улучшаются. Однако их возможности ограничены аппаратурными ресурсами кристалла и корпуса. Поэтому более широко распространены многокристалльные, а также многокристалльные секционные МП.

Для получения многокристалльного МП необходимо провести разбиение его логической структуры на функционально законченные части и реализовать их в виде БИС (СБИС). Функциональная законченность БИС многокристалльного МП означает, что его части выполняют заранее определенные функции и могут работать автономно, а для построения развитого процессора не требуется организации большого количества новых связей и каких-либо других электронных ИС (БИС). (Типичный пример - МПК БИС серии К581).

На рис. 4,а показано функциональное разбиение структуры МП при создании трехкристалльного микропроцессора (пунктирные линии), содержащие БИС операционного (ОП), БИС управляющего (УП) и БИС интерфейсного (ИП) процессоров.

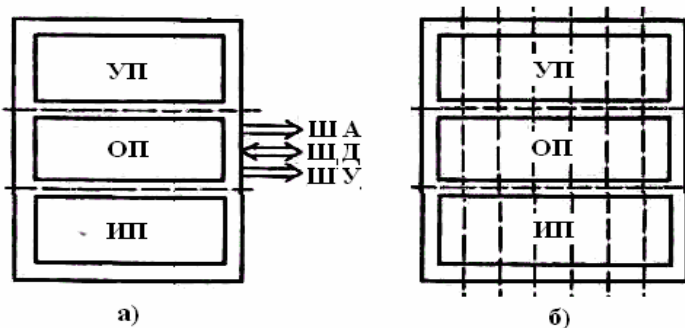


Рис. 4

Операционный процессор ОП служит для обработки данных, управляющий процессор УП выполняет функции выборки, декодирования и вычисления адресов операндов и также генери-

рует последовательности микрокоманд. Автономность работы и большое быстродействие БИС УП позволяет выбирать команды из памяти с большей скоростью, чем скорость их исполнения БИС ОП. При этом в УП образуется очередь еще не исполненных команд, а также заранее подготавливаются те данные, которые потребуются ОП в следующих циклах работы.

Такая опережающая выборка команд экономит время ОП на ожидание операндов, необходимых для выполнения команд программ. Интерфейсный процессор ИП позволяет подключить память и периферийные средства к МП; по существу, является сложным контроллером для устройств ввода — вывода информации. БИС ИП выполняет также функции канала прямого доступа к памяти.

Выбираемые из памяти команды распознаются и выполняются каждой частью МП автономно, и поэтому может быть обеспечен режим одновременной работы всех БИС МП, т. е. конвейерный поточный режим исполнения последовательности команд программы (выполнение последовательности с небольшим временным сдвигом). Такой режим работы значительно повышает его производительность.

Многокристальные секционные МП получают в том случае, когда в виде БИС реализуются части (секции) логической структуры процессора при функциональном разбиении ее вертикальными плоскостями (рис.4,6). Для построения многоразрядных МП при параллельном включении секций БИС МП в них добавляются средства «стыковки».

Для создания высокопроизводительных многоразрядных МП требуется столь много аппаратурных средств, не реализуемых в доступных БИС, что может возникнуть необходимость еще в функциональном разбиении структуры МП горизонтальными плоскостями. В результате рассмотренного функционального разделения структуры МП на функционально и конструктивно законченные части создаются условия реализации каждой из них в виде БИС. Все они образуют комплект секционных БИС МП.

Таким образом, микропроцессорная секция — это БИС, предназначенная для обработки нескольких разрядов данных или выполнения определенных управляющих операций. Секционность

БИС МП определяет возможность «наращивания» разрядности обрабатываемых данных или усложнения устройств управления микропроцессором при «параллельном» включении большего числа БИС.

С момента создания однокристалльные МП развились от простых специализированных 4-разрядных до 32-разрядных процессоров. Трехкристалльные МП имеют разрядность до 32 бит и параметры, сравнимые с параметрами старших моделей рядов мини-ЭВМ и средних ЭВМ общего применения.

Многокристалльные секционные МП имеют разрядность от 2—4 до 8—16 бит и позволяют создавать разнообразные высокопроизводительные процессоры ЭВМ.

Однокристалльные и трехкристалльные БИС МП, как правило, изготавливают на основе микроэлектронных технологий униполярных полупроводниковых приборов, а многокристалльные секционные БИС МП — на основе технологии биполярных полупроводниковых приборов.

Использование многокристалльных микропроцессорных высокоскоростных биполярных БИС, имеющих функциональную законченность при малой физической разрядности обрабатываемых данных и монтируемых в корпус с большим числом выводов, позволяет организовать разветвление связи в процессоре, а также осуществить конвейерные принципы обработки информации для повышения его производительности.

По структурному признаку различают МП с фиксированной разрядностью и МП с наращиваемой разрядностью (секционные МП).

МП с фиксированной разрядностью имеют строго определенную разрядность обрабатываемых слов, величина которой определяется разрядностью МП. МП с наращиваемой разрядностью позволяют на их основе секциями увеличивать число разрядов МПС до требуемой величины, что, как правило, используется при построении миниЭВМ и больших ЭВМ вычислительного типа.

По виду алгоритма работы управляющего устройства МП подразделяют на два вида:

- МП с жестким алгоритмом управления, реализуемым схемно (МП с фиксированным набором команд),

- МП с алгоритмом управления, реализуемым программным путем в виде последовательности микроопераций (МП с микропрограммным управлением). Здесь система команд определена не жестко, а зависит от микропрограммы, записанной в ПЗУ, входящей в состав устройства управления. Использование микропрограммного управления дает возможность получить необходимый набор команд, например, для воспроизведения (эмуляции) набора команд другого МП.

По разрядности обрабатываемой информации МП могут быть 4, 8, 12, 16, 24, 32 - разрядными. На практике наибольшее распространение имеют 32 - разрядные МП (Pentium, Celeron, AMD). Все большее применение находят 64-разрядные МП фирмы AMD.

По характеру временной организации работы МП делятся на синхронные и асинхронные.

В синхронных МП начало и конец выполнения каждой операции задаются устройством управления, то есть фаза начала и конца выполнения команды строго привязана к временной оси.

В асинхронных МП начало выполнения следующей операции начинается сразу же после окончания выполнения предыдущей операции.

По количеству одновременно выполняемых программ различают одно- и многопрограммные МП.

В однопрограммных МП на текущий момент времени выполняется только одна программа. Переход к выполнению другой программы происходит либо по завершению этой программы, либо по специальной команде условного или безусловного перехода, либо по прерыванию.

В многопрограммных МП одновременно может выполняться несколько программ, то есть обеспечивается мультипрограммный режим работы системы.

По виду технологии изготовления разрабатываются и выпускаются БИС МП:

- по униполярной технологии - р - каналные (р - МОП), n - каналные (n - МОП) и комплиментарные (КМОП) БИС;

- по биполярной технологии - БИС на базе транзисторно-транзисторной логики (ТТЛ), в том числе и с диодами Шотки

(ТТЛШ);

- по эмиттерно-связанной логики (ЭСЛ);
- по интегральной инжекционной логики (И²Л).

Вид технологии изготовления БИС во многом определяет степень интеграции микросхем, быстродействие, энергопотребление, помехозащищенность и стоимость МП. По комплексу этих признаков можно отдать предпочтение МП, выполненным по n - МОП и КМОП технологиям, обеспечивающих высокую плотность компоновки, высокое быстродействие и относительно малую стоимость. ЭСЛ и ТТЛШ технологии обеспечивают МП самое высокое быстродействие, но микропроцессорные БИС (МП БИС) при этом отличаются самой низкой плотностью компоновки и высоким энергопотреблением. МП на основе И²Л технологии обладают усредненными характеристиками. По плотности компоновки они уступают n - МОП, по быстродействию - ЭСЛ и ТТЛШ, а по стоимости - n - МОП и p - МОП МП. Вместе с тем, p - МОП технология обеспечивает МП наиболее низкую стоимость, но его быстродействие при этом является также наиболее низким.

В процессе развития микропроцессорных средств, кроме микропроцессорных БИС, были разработаны различные интегральные микросхемы, выполняющие различные функции и позволяющие в совокупности построить микроЭВМ требуемой структуры. Эти микросхемы совместно с МП БИС образуют микропроцессорный комплект (МПК БИС), который может быть определен как совокупность конструктивно и электрически совместимых интегральных схем, предназначенных для построения МП, микроЭВМ и других вычислительных устройств с определенным составом и требуемыми технологическими характеристиками.

Основу любого МПК БИС образует базовый комплект интегральных микросхем, который предназначен для построения МПС и может состоять из БИС однокристалльного или из нескольких корпусов многокристалльного МП. Базовый комплект, как правило, дополняется другими типами интегральных схем, на которых реализуются запоминающие устройства, устройства сопряжения с объектом и различные устройства ввода - вывода. Эти микросхемы в общем случае могут иметь другой номер серии или даже иной тип корпуса.

Минимальный набор микросхем из состава МПК БИС, позволяющих построить конкретный тип вычислительного устройства, называется микропроцессорным набором интегральных схем.

5. Особенности архитектуры 32-разрядных МП

5.1. Микропроцессоры с RISC – архитектурой

В развитии архитектур МП наблюдается два подхода. Первый из них относится к более ранним моделям процессоров и носит название МП с CISC (Complete Instruction Set Computer) архитектурой - процессоры с полным набором инструкций. К ним относится семейство процессоров 80×86 . Состав и назначение их регистров существенно неоднородны, широкий набор команд усложняет декодирование инструкций, на что расходуются аппаратные ресурсы. Возрастает число тактов, необходимое для выполнения инструкций.

Процессоры 80×86 имеют весьма сложную систему команд, что еще довольно терпимо при использовании ее в 8 - и 16 – разрядных МП. В начале 80-х годов архитектура CISC стала серьезным препятствием на пути реализации идеи «один процессор в одном кристалле», поскольку для работы с «традиционным» расширенным списком команд требуется очень сложное устройство центрального управления (обычно - микропрограммное), занимающее до 60% всей площади кристалла.

В процессорах семейства 80×86 , начиная с i80486, применяется комбинированная архитектура - CISC-процессор имеет RISC-ядро. Архитектура RISC (Reduced Instruction Set Computer — компьютер с сокращенным набором инструкций) была впервые реализована в 1979 г. в миникомпьютере IBM801. В ней воплотились три основных принципа:

- ориентация системы на поддержку языка высокого уровня с помощью развитого компилятора;
- использование примитивного набора инструкций, который полностью реализуется аппаратными средствами;
- организация памяти и ввода—вывода, которая позволяет выполнять процессором большинство инструкций за один такт.

Первые микропроцессоры с архитектурой RISC были разработаны и изготовлены в начале 80-х годов в Калифорнийском (г. Беркли) и Стэнфордском университетах. Разработчики этих МП ставили перед собой задачу достижения наивысшей производительности при наименьшей сложности. В ходе ее решения сложились два подхода.

Первый заключается в снижении числа обращений в память за счет увеличения емкости регистрового файла и организации его в виде перекрывающихся регистровых окон. Архитектура, созданная на этой основе, была впервые реализована в МП RISC I, разработанном в г. Беркли (берклийская архитектура).

Другой подход заключается в устранении задержек конвейера за счет переупорядочения инструкций и интенсивного использования регистров МП при помощи оптимизирующего компилятора. Архитектура, реализующая этот способ, была разработана и впервые применена в г. Стэнфорде (станфордская архитектура).

5.1.1. Общие принципы построения

Рабочие станции и серверы, созданные на базе концепции RISC, завоевали лидирующие позиции благодаря своим исключительным характеристикам. Дело дошло до предсказаний скорого отмирания более традиционных CISC-систем. Чисто академический интерес середины 80-х годов к архитектуре RISC в начале 90-х годов сменился бурным ростом производства промышленных RISC-систем. Практически все ведущие производители - IBM, Hewlett-Packard, DEC, Silicon Graphics - создали процессоры с RISC-архитектурой и выпустили на рынок новые семейства рабочих станций и серверов на их базе. Более того, RISC-системы вышли за границы узких профессиональных приложений и находят все большее признание среди средних пользователей.

В теории цифровых логических систем есть известная аксиома, которая гласит, что любой компьютер в принципе может быть построен с использованием всего одного типа элементов - вентиля "И - НЕ / ИЛИ - НЕ". Однако никому из разработчиков машин 60-х и 70-х годов не приходило в голову отказаться от каталога из десятков и сотен логических микросхем и спроектиро-

вать компьютер на одном типе вентиля.

Никому, кроме Сеймура Крея. Результат известен: суперкомпьютер CRAY-1, созданный в рекордно короткие сроки, оказался меньше и быстрее всех своих предшественников. Нечто подобное произошло и в процессе становления RISC-архитектуры. Идея, заложенная в основу RISC-архитектуры, состояла в следующем: оставить в системе команд всего несколько десятков наиболее употребимых и наиболее универсальных инструкций, исключив сложные и редко используемые.

Результатом должно было стать существенное упрощение центрального управления, а значит, высвобождение части поверхности кристалла процессора для размещения более мощных средств обработки данных. Так возникла философия RISC-архитектуры – «меньше команд - выше скорость», которая основывается на двух фундаментальных постулатах:

- скорость компьютерной обработки определяется не столько быстродействием аппаратных средств, сколько хорошим взаимодействием программного обеспечения и аппаратуры
- за скорость всегда надо платить усложнением либо аппаратуры, либо программных средств, либо того и другого.

Их реализация давно интересовала разработчиков МПС. Еще задолго до разделения компьютеров на RISC- и CISC-семейства было освоено два способа повышения скорости вычислений – «быстрые» технологии и параллелизм обработки. На пути ускорения обработки данных в принципе хорошо известны: схемы на арсениде галлия примерно в четыре раза производительнее схем на кремниевой основе, насыщенная логика по быстродействию уступает, оптимизация откомпилированного кода теоретически позволяет в 2 - 4 раза сократить время выполнения программы и т. д.

Концепция RISC - архитектуры базируется на почти очевидной логической формуле: если «быстрые» технологии и параллельная обработка для всего списка команд недостижимы из-за высокого уровня затрат, то надо ускорять только часто выполняемые операции, а редко применяемыми и сложными следует пожертвовать ради повышения общей производительности. Заметная разница между RISC-компьютерами 80-х и 90-х годов и CISC- ма-

шинами 60-х годов заключается в числе аппаратных шагов, приходящихся на инструкцию. В RISC - процессоре одна инструкция выполняется за один шаг, тогда как в CISC та же инструкция может вызвать сотни и тысячи аппаратных действий.

Конечно, программирование с помощью подобных насыщенных операций позволяет получить компактный исполняемый модуль, но возникает естественный вопрос: «Что лучше - короткая программа с медленными инструкциями или длинная программа с быстрыми инструкциями?». Ответ на него помогли дать прикладные исследования в лабораториях фирмы IBM. Было сконструировано подмножество языка PL/1 под названием PL/8 и написан компилятор с оптимизацией кода для гипотетического компьютера, система команд которого использовала короткие инструкции типа «регистр-регистр». Имитация работы этого компьютера проводилась на мэйнфрейме IBM/370 модели 168. Этот эксперимент дал весьма впечатляющий результат: С большинством наиболее часто употребляемых программных операторов компьютер справилась в 2 - 3 раза быстрее, чем IBM 370/168, запрограммированная при помощи стандартного варианта языка PL/1.

Для последующих семейств мэйнфреймов производства IBM данное соотношение несколько уменьшилось за счет конвейеризации процессора и большего объема кэш-памяти, однако принципиальный вывод из эксперимента в IBM не потерял своей значимости: отказ от применения редких инструкций и оптимизация использования регистров ускоряют вычислительный процесс более чем в два раза.

Итак, исключение из системы команд редко применяемых инструкций и ориентация аппаратных и программных средств на операции типа «регистр-регистр» открывают широкие возможности для экономии оборудования без существенной потери производительности. Но как только на кристалле процессора оказалось свободное место, сразу же нашлись желающие занять его под более мощные средства обработки. Так, в 1985 г. фирма Acorn Corporation of England выпустила 32-разрядный RISC-процессор ARM, примерно эквивалентный по степени интеграции 8-разрядному CISC-процессору Intel 8080 (около 25 тыс. транзисторов), но со значительно большим быстродействием.

Правда, произошло это уже после того, как были сформулированы основные законы RISC-архитектуры. Законы RISC в самом начале 80-х годов почти одновременно завершились теоретические исследования в области RISC-архитектуры, проводившиеся в Калифорнийском университете (г. Беркли), Станфордском университете и в корпорации IBM. Именно тогда были сформулированы четыре основных принципа RISC-архитектуры:

1. Каждая команда независимо от ее типа выполняется за один машинный цикл, длительность которого обратно пропорциональна тактовой частоте процессора и должна быть максимально короткой. Стандартом для RISC - процессоров считается длительность машинного цикла, равная времени сложения двух целых чисел (для современного уровня развития технологии эта величина составляет от 3 до 10 нс).

2. Все команды должны иметь одинаковую длину и использовать минимум адресных форматов; это резко упрощает логику центрального управления процессором. Другим важным следствием принципа простоты адресации является то, что RISC - процессор способен выбирать очередную команду в темпе обработки, т. е. одну команду за один цикл.

3. Обращение к памяти происходит только при выполнении операций записи и чтения, модификация операндов в памяти возможна лишь с помощью команды «запись», вся обработка данных осуществляется исключительно в регистровой структуре процессора.

4. Система команд должна обеспечивать поддержку языков высокого уровня. Имеется в виду подбор системы команд, наиболее эффективной для различных языков программирования.

Само собой разумеется, что четыре перечисленных базовых принципа RISC-архитектуры не существуют вне основного закона RISC: система команд должна содержать минимум наиболее часто используемых и наиболее простых инструкций.

Конечно, в компьютерной практике можно найти немало примеров широкого толкования принципов RISC, однако один закон RISC-архитектуры соблюдается всеми разработчиками неукоснительно - обработка данных должна вестись только в рамках регистровой структуры и только в формате команд «регистр- ре-

гистр».

Регистры - основное достоинство и главная проблема RISC. Все существующие RISC-процессоры базируются на единственном типе обработки данных в формате «регистр-регистр», а точнее, «регистр-регистр-регистр»: $R1:=R2,R3$. Это позволяет без существенных затрат времени выбрать операнды из адресуемых оперативных регистров и записать в регистр результат операции.

Кроме того, трехместные операции дают компилятору большую гибкость по сравнению с типовыми двухместными операциями формата «регистр-память» архитектуры CISC. В сочетании с быстродействующей арифметикой RISC-операции типа «регистр-регистр» становятся очень мощным средством повышения производительности процессора. Проблема заключается в том, что в процессе выполнения задачи RISC-система неоднократно вынуждена обновлять содержимое регистров процессора, причем за минимальное время, чтобы не вызвать длительных простоев арифметического устройства (а это прямые потери производительности). Для CISC-систем подобной проблемы не существует, поскольку модификация регистров может происходить на фоне обработки команд формата «память – память».

Существует два подхода к решению проблемы модификации регистров в RISC - архитектуре: аппаратный, предложенный в проектах RISC-1и RISC-2 университета в Беркли, и программный, разработанный специалистами IBM и Станфордского университета. Принципиальная разница между ними заключается в том, что аппаратное решение основано на стремлении уменьшить время вызова процедур за счет установки дополнительного оборудования процессора, тогда как программное базируется на возможностях компилятора и является более экономичным с точки зрения аппаратуры центрального процессора.

В RISC-архитектуре используется механизм переключения множественных перекрывающихся регистровых окон - MORS (Multiple Overlapping RegisterSets), иногда называемый структурой регистрового файла Rolodex. Механизм MORS послужил основой архитектуры RISC-1, в соответствии с которой процессор содержит 138 регистров для хранения данных. Из них десять, именуемых глобальными, всегда «видны» программе; их основное назна-

чение - хранение данных, являющихся общими для всех процессов в текущем контексте программы. Остальные 128 регистров разбиты на восемь перекрывающихся окон по 22 регистра. В каждый момент времени программа, исполняемая на RISC-1, «наблюдает» десять глобальных регистров и одно целое окно, т. е. всего 32 регистра.

Идея структуры MORS заключается в минимизации затрат процессорного времени при обращении к процедурам. Для этого каждое из восьми окон связано с конкретной процедурой, а регистры окна разделены на верхние, локальные и нижние. При вызове процедуры В из процедуры А активное окно регистравого файла смещается на шесть позиций так, что верхние регистры процедуры А перекрываются нижними регистрами процедуры В.

Перекрывающиеся зоны окон - это физически одни и те же регистры, доступные обеим процедурам. Они используются для передачи параметров, адресов возврата и позволяют обращаться к процедуре, не обмениваясь данными с оперативной памятью. Таким образом, вызов процедуры реализуется не сложнее, чем, скажем, суммирование регистровых операндов. К тому же эта операция выполняется практически моментально: для обращения к процедуре или для возврата в точку вызова достаточно переместить указатель активного окна регистравого файла. В этом состоит важнейшая особенность архитектуры RISC-1.

Вероятно, именно благодаря своей логической стройности архитектура RISC-1 послужила основой для разработки массовых процессоров Pyramid и SPARC, правда, с небольшими изменениями в организации регистравого файла Rolodex (в SPARC программа «видит» окно из тех же 32 регистров, но количество глобальных, верхних, локальных и нижних регистров одинаково - по восемь в каждой зоне).

Однако структура MORS обладает двумя недостатками - оптимальное размещение процедур по окнам регистравого файла является далеко не тривиальной задачей для ОС, а выбранное число из восьми процедур, сохраняемых в регистравом файле, представляется, скорее, эмпирическим значением. Во всяком случае можно найти множество примеров, когда задача включает существенно большее количество процедур и при этом возникает реаль-

ная проблема модификации одного или нескольких окон для активизации процедур, сохраняемых в оперативной памяти.

В компьютере RISC-1 ситуация, когда требуется выполнить вызов очередной вложенной процедуры, а все окна регистрового файла заняты, разрешается с помощью логики процессора, которая формирует специальную программную ситуацию. При этом процессор инициирует программу ОС, высвобождающую одно или несколько регистровых окон, т. е. передает (trap) содержимое регистров в оперативную память. В случае применения RISC-1 в качестве машины общего назначения такое решение казалось весьма приемлемым, поскольку обычно ситуация trap возникает в одном из ста обращений к процедуре.

Но для работы в реальном времени один процент случаев оказывается недопустимо большой величиной. Действительно, если прерывание происходит в момент, когда все регистровые окна заняты, то инициируется выполнение процедуры trap и время реакции становится недетерминированным - ситуация, крайне опасная для систем реального времени. Если к тому же потребуется контекстное переключение от одной задачи к другой, то придется передать в память от одного до восьми окон в зависимости от текущего состояния прерываемой программы. А это, в свою очередь, означает, что время контекстного переключения будет изменяться в широких пределах: от 60 до 840 машинных циклов (прерывание, выполняемое внутри регистрового файла RISC-1, занимает не более трех циклов).

Именно такой разброс и является неприемлемым для систем реального времени, в которых период реакции должен быть строго детерминированным. Попытки решения данной проблемы привели к совершенствованию процедуры trap в проекте RISC-2 университета в Берклии в проекте Omega. Существенное отличие названных проектов от RISC-1 состоит во включении в архитектуру компьютера динамического механизма быстрого сохранения регистровых окон в специальной памяти.

5.1.2. Берклийская архитектура

Согласно статистике, 50—70% используемых операндов

составляют локальные переменные и параметры процедур. Их размещение в регистровом файле МП позволяет существенно снизить число обращения в память. В RISC МП с берклийской архитектурой регистры группируются в несколько банков, чтобы для каждой процедуры процессор мог назначить свой набор регистров, переключение которого осуществляется модификацией аппаратного указателя.

Рассмотрим организацию регистрового файла (табл. 1) на примере МП RISC II, разделенного на виртуальные регистровые окна емкостью 32 регистра каждое (рис.5).

Таблица 1

Параметры	RISC I	RISC II	SOAR	SPARC	Am29000
Число вирт. рег. окон	6	8	8	7	8
Число локальных рег. в вирт. окне	14	22	16	24	16
Глубина перекрытия вирт. рег. окон	4	6	8	8	0
Число глоб. регистров	18	10	8	8	64
Общее число рег. в рег. файле	78	138	72	120	192

Регистры 26—31 (верхние) содержат параметры, переданные от вызывающей процедуры. Регистры 16—25 (локальные) используются для хранения локальных скалярных переменных, 10—15 (нижние) — для хранения переменных и параметров, передаваемых вызываемой процедуре. Регистры 0—9 предназначены для хранения глобальных переменных.

Таким образом, возможно обращение каждой процедуры к 32 регистрам. Соседние регистровые банки, используемые вызывающей и вызываемой процедурами, перекрываются, так что параметры могут быть переданы процедуре без какого-либо перемещения данных. На каждое обращение к процедуре назначается новый набор регистров R10...R31. При этом нижние регистры вызывающей процедуры становятся верхними вызываемой, поскольку они совмещены физически.

Таким образом, без перемещения информации параметры,

хранящиеся в регистрах 10—15 вызывающей процедуры, появляются в регистрах 26—31 вызываемой. Рис. 2 иллюстрирует этот подход для случая, когда процедура А вызывает процедуру В, а та, в свою очередь, вызывает С.

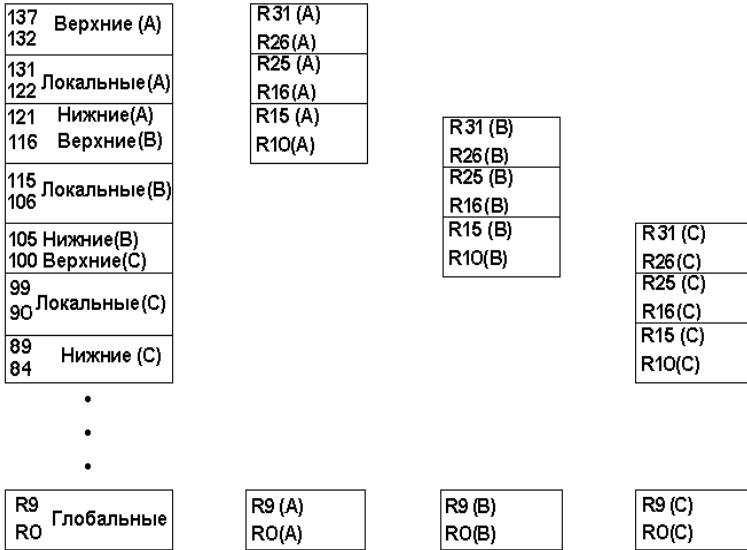


Рис. 5

Во многих программах глубина вложенности процедур превышает число регистровых окон, поэтому в МП с берклийской архитектурой освобождение регистровых банков при переполнении регистрового файла осуществляется путем передачи их содержимого в память. Если глубина вложенности выходит за пределы логических наборов регистров, содержащихся в регистровом файле, то начинается программная или аппаратная обработка прерывания. Содержимое нескольких регистров передается в память, в отдельной области которой организован стек переполнения регистров. Сигналы переполнения/недополнения перемещают указатель стека на его вершину.

Эффективность такой организации обращения к процедурам зависит от частоты появления переполнений/недополнений, в

большей степени связанной с логическими изменениями глубины стека, нежели с его абсолютной глубиной.

Исследования показали, что при восьми регистровых банках переполнения/недополнения возникают менее чем в 1% обращений к процедурам. Для того, чтобы к переменным в регистрах можно было обращаться с помощью указателей, все регистры отображаются в адресное пространство регулярной памяти.

5.1.3. Станфордская архитектура

Скорость обработки инструкций в МП с конвейерной архитектурой существенно снижается из-за возникновения конфликтных ситуаций следующего типа:

- программа осуществляет переход, для которого требуется очистка конвейера и загрузка его новыми инструкциями (зависимость по адресу);
- инструкции запрашивают информацию, которая еще не получена от обрабатываемых в конвейере инструкций (зависимость по данным);
- инструкциям в конвейере одновременно требуются обращения к одному и тому же ресурсу — шине памяти, регистру или АЛУ.

Возможны различные случаи задержки конвейера, содержащего пять ступеней обработки инструкции (рис. 6): выборку инструкции (IF), дешифрацию инструкции (ID), выборку операнда (OF), вычисление (OE) и запоминание результата (OS). В первом случае (см. рис. 6,а) выполняется инструкция безусловного перехода JMP, что задерживает конвейер до тех пор, пока не завершится ее обработка, включающая четыре этапа:

- дешифрацию инструкции JMP,
- выборку операнда из программного счетчика РС,
- модификацию содержимого РС,
- запись результата обратно в программный счетчик; задержка в этом случае длится четыре такта.

Во втором случае (рис. 6,б) инструкция INC A не может выбрать операнд A, пока предыдущая инструкция ADD B, C, A не завершит запись результата операции в регистр A; задержка длит-

ся два такта.

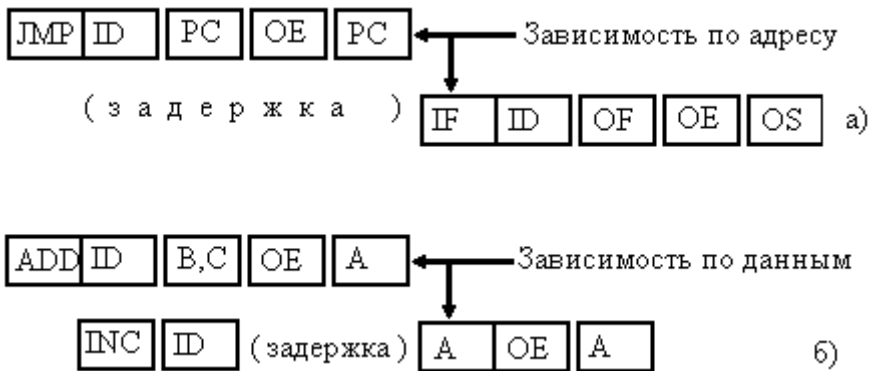


Рис. 6

Такие конфликтные ситуации - зависимость по адресу и по данным, свойственны всем МП с конвейерной обработкой инструкций. Традиционно эти проблемы решаются аппаратно за счет опережающей выборки инструкций в точке перехода или за счет блокировки конвейера в случае возникновения конфликтной ситуации.

Станфордская архитектура предусматривает устранение задержек конвейера при помощи оптимизирующего компилятора, переупорядочивающего инструкции так, чтобы они не зависели одна от другой при обработке их в конвейере. При этом вводится инструкция задержанного перехода (табл. 2), которая применяется также и в некоторых RISC МП с берклийской архитектурой. Задержанный переход выполняется так, что инструкция, следующая за инструкцией перехода, выполняется до передачи управления в точку перехода. В это время процессор имеет возможность выбрать инструкцию по адресу перехода и загрузить ее в конвейер.

Традиционные МП реализуют этот фрагмент программы как обычный переход - инструкции выполняются в последовательности: 100, 101, 102, 105 и т.д. Чтобы получить такой эффект в RISC МП, необходимо ввести в задержанный переход инструкцию NOP, не выполняющую операций. Тогда последовательность станет такой: 100, 101, 102, 103, 106 и т.д.; обрабатывается она опти-

мизирующим компилятором, который по возможности переупорядочивает ее так чтобы максимально использовать цикл после задержанного перехода. Последовательность исполнения инструкций принимает вид: 100, 101, 102, 105 и т.д. Поскольку инструкция, следующая за инструкцией перехода ,выполняется всегда и переход по адресу 101 не зависит от выполнения инструкции ADD по адресу 102, последовательность эта эквивалентна исходному фрагменту программы.

Таблица 2

Адрес	Обычный переход	Задержанный переход	Оптимиз. задержанный переход
100	LOAD X,A	LOAD X,A	LOAD X,A
101	ADD 1,A	ADD 1,A	JMP 105
102	JMP 105	JMP 106	ADD 1,A
103	ADD A,B	NOP	ADD A,B
104	SUB C,B	ADD A,B	SUB C,B
105	STORE A,Z	SUB C,B	STORE A,Z
106		STORE A,Z	

Концепция задержанного перехода применяется в ряде случаев для выполнения инструкций обращения в память за один такт. Непосредственный способ выполнения этих инструкций требует двух тактов: один необходим для вычисления адреса, второй — собственно для обращения в память. Модифицированные инструкции обращения в память называются задержанными загрузками и предполагают некоторые изменения аппаратуры: память и регистры МП должны быть оснащены дополнительными портами. Зависимость по данным, которая может возникнуть при выполнении задержанной загрузки, устраняется при помощи оптимизирующего компилятора.

Для RISC МП со станфордской архитектурой характерно небольшое число регистров общего назначения, используемых для хранения локальных переменных и параметров проце-

дур. Распределение регистров между переменными выполняется оптимизирующим компилятором.

5.1.4. Применение RISC - архитектуры в 32-разрядных МП

Каждой из рассмотренных разновидностей архитектуры присущи как достоинства, так и недостатки, поэтому обе они получили примерно одинаковое развитие и распространение.

Первые МП с архитектурой RISC (RISC I и RISC II) были разработаны и изготовлены в начале 80-х годов в Калифорнийском университете (г. Беркли). Они имели небольшой набор инструкций и простые режимы адресации. Все арифметические и логические инструкции выполнялись над регистровыми операндами, и только две — LOAD и STORE — использовались для обращения в память. На площади кристалла, освободившейся от схемы дешифрации сложных команд и формирователя последовательности микрокоманд, присущих традиционной архитектуре МП, размещен большой регистровый файл. Организация его в виде перекрывающихся регистровых окон позволяла каждой процедуре назначать новый набор регистров и передавать данные от одной процедуры к другой простым изменением аппаратного указателя. Регистровые окна, впервые примененные в берклийских МП обеспечивают эффективную поддержку вложенных процедур, сокращают число обращений в память и значительно повышают производительность МП.

В 1983 г. в Станфордском университете был разработан и изготовлен МП MIPS (Microprocessor without Interlocked Pipeline Stages - микропроцессор без блокировок конвейера). Целью проекта MIPS было создание МП, свободного от задержек конвейера, связанных: с выполнением переходов, зависимостью инструкций по данным и одновременным обращением нескольких инструкций, обрабатываемых в конвейере, к одному ресурсу. Благодаря разработке и применению оптимизирующего компилятора в МП MIPS удалось преодолеть недостатки берклийской архитектуры, обусловленные, прежде всего, необходимостью использования большого регистрового файла, занимающего до 60% площади кристалла.

Архитектура МП SOAR (Smalltalk On A RISC) является развитием берклийской ветви RISC. Он предназначен для поддержки языка высокого уровня Smalltalk-80 и функционирования в составе рабочей станции Sun. SOAR поддерживает два типа данных: 31-разрядные переменные со знаком и 28-разрядные указатели. Каждое слово данных содержит признак типа (tag), который проверяется одновременно с выполнением арифметических операций. Если оба операнда оказываются целыми, то через один такт получается правильный результат. В противном случае SOAR обращается к программам обработки указателей. Другой особенностью его является использование инструкций групповой загрузки—хранения, позволяющих передавать содержимое восьми регистров за девять тактов вместо шестнадцати (один такт тратится на выборку инструкции и восемь — на обращения в память).

МП ARM (Acorn RISC Machine) разработан специально для персонального компьютера модели В фирмы Acorn Computers и воплощает основные принципы станфордской архитектуры. Он обладает небольшим набором аппаратно-реализованных инструкций, имеющих одинаковый 32-разрядный формат. Для обеспечения быстрой реакции на прерывания разработчики ARM исключили из его набора команд длинные операции, которые невозможно прервать. Как и МП SOAR, ARM поддерживает инструкции групповой загрузки—хранения, позволяющие повысить скорость обмена между МП и памятью. В отличие от других RISC МП, в ARM не используются задержанные переходы. Для устранения задержек конвейера при выполнении переходов в каждую инструкцию введено 4-разрядное поле кода условия; выполняется оно только в том случае, если установлен код соответствующего условия.

МП ROMP (Research/Office division Microprocessor) - микропроцессор отделения по автоматизации научных и Учрежденческих работ) корпорации IBM. Он предназначен для работы в качестве центрального процессора в персональном компьютере RT PC (RISC Technology Personal Computer), реализует станфордскую архитектуру и оснащен развитым оптимизирующим компилятором.

Особенностью ROMP является длинное 40 - разрядное адресное слово и соответствующее ему адресное пространство 1Т

байт.

При разработке ROMP большое внимание уделялось повышению скорости обмена между процессором и памятью. Для этого используется аппаратно управляемый буфер преобразованных адресов, применявшийся в ранних моделях IBM. Он выполняет эффективное преобразование виртуальных адресов в физические. Кроме того, каждое слово данных в ROMP снабжается тегом, по которому определяется получатель данных, поэтому любой элемент данных может перебраться, как только он будет готов.

Центральный процессорный элемент (ЦПЭ) R2000 фирмы Mips Computer Systems составляет основу для построения супермини компьютеров этой фирмы. Он имеет станфордскую архитектуру, работает в программной среде ОС Umips, совместимой с ОС Unix, языков высокого уровня Си, Фортран-77 и Паскаль. Для управления внешней кэш-памятью используется буфер преобразованных адресов. R2000 работает в составе трехкристального набора, в который входят также сопроцессор для операций с плавающей точкой и буфер записи, предназначенный для согласования скоростей работы ЦПЭ и динамической памяти.

ЦПЭ MD 484 фирмы McDonnell Douglas изготавливается по GaAs-технологии и имеет расчетную производительность 100 млн. инструкций в 1 с. Благодаря станфордской архитектуре, MD 484 содержит небольшое число компонентов (23,2 тыс. транзисторов и 10,4 тыс. резисторов) и поддерживает набор инструкций МП MIPS. Он имеет развитую систему прерываний, включающую прерывания из-за неправильной адресации, внутреннего переполнения и системных обращений. Для передачи инструкций и данных используются три двунаправленные 32-разрядные шины, каждая линия которых заряжается с помощью большого п-канального транзистора. Высокая производительность ЦПЭ обеспечивается за счет сопроцессора для операций с плавающей точкой и устройства управления памятью, входящих в МП набор.

МП SPARC (Scalable Processor ARChitecture) фирмы Sun Microsystems воплощает основные принципы берклийской архитектуры: он содержит большой регистровый файл емкостью 120 32-разрядных регистров и использует задержанные переходы. МП

реализован на вентиляционной матрице фирмы Fujitsu, выполненной по 1,5 мкм КМОП-технологии и содержащей 20 тыс. вентиляей. МП SPARC предназначен для работы в программной среде ОС Unix в составе рабочей станции Sun-4.

МП Am29000 фирмы AMD отличается наивысшей производительностью среди серийных RISC МП и реализует усовершенствованную версию берклийской архитектуры. В него входит регистровый файл емкостью 192 регистра, разделенный на банки по 16 регистров; использует он задержанные переходы для устранения задержек конвейера, вызванных зависимостью по адресу, и выполняет команды обращения в память за один такт благодаря применению концепции задержанной загрузки. Кроме того, для ускорения обмена между МП и внешней памятью в Am29000 введен буфер преобразованных адресов.

Трехкристальный МП набор Clipper C100 корпорации Fairchild позволяет реализовать суперЭВМ на одной плате. Архитектура Clipper C100 имеет некоторые черты RISC, например аппаратную реализацию большинства инструкций, однако ее нельзя отнести ни к одному из рассмотренных выше классов. Этот МП способен выполнять все арифметические операции с плавающей точкой по стандарту IEEE 754, он поддерживает десять типов данных и девять режимов адресации, а его система прерываний включает 256 векторных прерываний и 128 системных обращений.

Набор инструкций Clipper C100 содержит 101 аппаратно-реализованную инструкцию и 67 макроинструкций, выполняющих преобразование чисел из формата с фиксированной точкой в формат с плавающей точкой и обратно, обработку символьных строк, хранение - восстановление регистров, обработку прерываний и другие операции. Для поддержки макроинструкций используется микропрограммное ПЗУ емкостью 1К 48-разрядных слов. Вычислительный модуль Clipper C100 работает в программной среде ОС Unix и языков высокого уровня Си, Фортрая и Паскаль.

МП система MC88000 фирмы Motorola включает процессор MC88100 и два кристалла памяти MC88200.

MC88100 содержит регистровый файл небольшой емкости, блоки обработки чисел в формате с фиксированной и плавающей точкой, а также до шести заказных блоков специальных функций.

МП имеет четыре порта ввода - вывода и поэтому может быть использован для построения мультипроцессорных систем.

5.1.5. Особенности интеграции элементов RISC-архитектуры в процессорах серии x86

Организация первых моделей процессоров - i8086/8088 - была направлена, в частности, на сокращение объёма программ, отличавшихся малой оперативной памятью. Расширение спектра операций, реализуемых системой команд, позволило уменьшить размер программ, трудоёмкость их написания и отладки, но повысило трудоёмкость их разработки.

Последнее проявилось в удлинении сроков разработки CISC-процессоров и проявлении различных ошибок в их работе. Кроме того, нерегулярность потока команд ограничила развитие топологии временным параллелизмом обработки инструкций на конвейере «выборка команды, дешифрация команды, выборка данных, вычисление - запись результата».

Эти недостатки обусловили необходимость разработки альтернативной архитектуры, нацеленной, прежде всего, на снижение нерегулярности потока команд уменьшением их общего количества. Это было реализовано в RISC-процессорах.

Сокращение нерегулярности потока команд позволило обогатить топологию RISC-процессоров пространственным параллелизмом, специализированными аппаратными АЛУ, независимыми кэш данных и команд, отдельными шинами ввода-вывода. Последние, в частности, увеличили длину конвейеров команд. Всё это повысило и производительность - увеличением числа операций, выполняемых за один такт, и быстродействие - сокращением пути транзактов - RISC-процессоров. При этом срок разработки данных чипов свидетельствует о том, что её трудоёмкость меньше, чем в случае CISC-процессоров.

На мировых рынках CISC-процессоры представлены, в основном, клонами процессоров Intel серии x86, производимыми AMD, Cyrix, а RISC -чипами Alpha, PowerPC, SPARC. Уступая во многом последним, процессоры x86 сохранили лидерство на рынке персональных систем лишь благодаря совместимости про-

граммным обеспечением младших моделей. В свою очередь, достоинства RISC-процессоров укрепили их позиции на рынке высокопроизводительных машин.

Несмотря на формальное разделение «сфер влияния», между представителями этих архитектур в начале 90-х годов началась острая конкуренция за улучшение характеристик. В первую очередь, производительности и её отношения к трудоёмкости разработки процессоров.

Первыми к этому пришли разработчики Intel, реализовавшие в i80486 пространственный параллелизм вычислений с фиксированной и плавающей запятой. Поддержка каждого АЛУ своей шиной данных/команд и регистровым блоком повысила производительность i80486 одновременным выполнением указанных команд. Кроме того, интеграция кэш-памяти и очереди команд позволила поднять частоту ядра процессора в 2-3 раза в сравнении с системной шиной. Однако совместное размещение данных и команд ограничило эффективность кэш необходимостью его полной перезагрузки после выполнения команд переходов.

Для устранения недостатка в Pentium реализованы отдельные кэш для команд и данных, позволяющие после переходов перезагружать лишь команды- такое решение называется Гарвардской архитектурой, а также предсказание переходов, снижающее частоту перезагрузок. Последнее достигается предварительной загрузкой в кэш команд с обоих разветвлений. Введение второго целочисленного тракта, состоящего из АЛУ, адресного блока, шин данных/команд, и работающего на общий блок регистров, повысило производительность поддержкой параллельной обработки целочисленных данных. Развитием данной тенденции стало обогащение Pentium MMX мультимедийным трактом, образованным АЛУ, шинами данных/команд и регистровым файлом.

При этом в случае выборки двух целочисленных команд, зависящих по данным, каждая из них выполняется последовательно, что снижает эффективность работы процессора. Частично поправило ситуацию создание оптимизирующих рекомпиляторов, например, Pen-Opt фирмы Intel, разделяющих по возможности такие команды.

Реализация описанного управления обработкой команд

CISC-формата вызвала дополнительный рост трудоёмкости разработки Pentium в сравнении с i8086/i80486, что привело не только к увеличению её реального срока на 27% в сравнении с ожидаемым, но и к проявлению ошибок в первых моделях данного процессора

Учтя это, компания Intel реализовала в Pentium Pro RISC-подобную организацию вычислений. Интерпретация команд 80×86 внутренними - RISC86 - инструкциями VLIW-формата помимо снижения нерегулярности их потока, обеспечила синхронную загрузку четырёх операционных - по два с плавающей и фиксированной запятой - АЛУ этого чипа. Термин VLIW расшифровывается как "очень длинное командное слово" (Very Long Instruction Word). Инструкции этого формата содержат команды для всех параллельных АЛУ.

Обогащение управления обработкой предвыборкой данных и команд, предполагаемых к обработке в ближайшие 20 тактов, повысило регулярность загрузки вычислительных трактов. В свою очередь, осуществление предвыборки из интегрированного на кристалле кэш второго уровня, обслуживаемого отдельными шинами «интерфейс-кэш» и «кэш-АЛУ» и работающего на частоте АЛУ, повысило быстродействие подготовки команд в сравнении с внешними кэш. Дополнительное повышение производительности Pentium Pro обеспечило увеличение длины команд до 11 ступеней введением ступеней трансляции и предвыборки. Кроме того, интеграция кэш второго уровня позволила умножить частоту ядра в 5-6 раз.

В архитектуре P6 RISC - решения впервые в семействе 80×86 перестали быть лишь дополнением исконных CISC - средств повышения производительности - роста разрядности, отложенной записи шины и других. Поэтому частица PRO в названии первого процессора этой серии обозначает «полноценная RISC-архитектура» (Precision RISC Organization).

Топологические новинки Pentium II - интеграция тракта MMX, мультипроцессорный интерфейс Xeon, вынесение кэш второго уровня на кристалл в корпусе чипа, как и полное устранение кэш второго уровня в Celeron, не имеют в данном случае качественной роли и направлены на оптимизацию отношения характери-

стик этих процессоров, к их цене.

Снижение трудоёмкости и длительности разработки аппаратно – программных реализаций алгоритмов работы Pentium Pro, позволило достигнуть роста характеристик сочетанием преимуществ RISC&CISC архитектур.

Сказанное иллюстрирует и организация современных RISC-процессоров. Их отличает, в данном случае, развитие систем команд с целью сохранения иерархической совместимости и снижения трудоёмкости разработки программ. Это сближает технологии обработки команд процессорами упомянутых архитектур. Например, SuperSparc взяли от последних моделей 80×86 предсказание переходов и предварительную интерпретацию кода.

Таким образом, развиваясь, каждая из рассмотренных архитектур, «отказавшись» от своих черт - CISC от скалярности вычислений, RISC от «простоты» системы команд, приобрела лучшие характеристики её представителей. Это подтверждает и процессор Merced, разработанный Intel и Hewlett Packard.

Имеющиеся сведения позволяют предположить, что его архитектура продолжит тенденции Pentium Pro по оптимизации обработки внутренних VLIW – подобных команд реализацией эффективных архитектурных решений при одновременной оптимизации преобразования «внешних» инструкций. Особо отмечаются намерения создания двух вариантов этого чипа, различающихся лишь множеством этих инструкций. Первый совместим с CISC-семейством 80×86, второй - с RISC-процессорами Alpha.

Merced в известной степени прекратил соперничество CISC и RISC, в ходе которого представители данных архитектур улучшили свои характеристики. Это позволило предположить, что дальнейшее развитие массовых процессоров пройдёт по пути развития топологических и микропрограммных решений вычислительного ядра RISC - организации при одновременном повышении возможностей CISC - подобной системы команд.

5.2. МП с традиционной архитектурой

5.2.1. Intel Pentium 4

Одной из особенностей архитектуры процессора Pentium 4 является гарвардская внутренняя структура, реализуемая путем разделения потоков команд и данных, поступающих от системной шины через блок внешнего интерфейса и размещённую на кристалле процессора общую кэш-память 2-го уровня L2 (рис. 7).

Блок внешнего интерфейса реализует обмен процессора с системной шиной, к которой подключается память, контроллеры ввода / вывода и другие активные устройства системы. Обмен по системной шине осуществляется с помощью 64-разрядной двунаправленной шины данных, 41-разрядной шины адреса (33 адресных линии и 8 линий выбора байтов), обеспечивающей адресацию до 64 Гб внешней памяти.

Архитектура МП является суперскалярной, что обеспечивает одновременное выполнение нескольких команд в параллельно работающих исполнительных устройствах. Суперскалярность архитектуры реализуется путем организации исполнительного ядра процессора в виде ряда параллельно работающих блоков. Арифметико-логические блоки ALU производят обработку целочисленных операндов, которые поступают из заданных регистров блока регистров замещения (БРЗ). В эти же регистры заносится и результат операции. При этом проверяются также условия ветвления для команд условных переходов и выдаются сигналы перезагрузки конвейера команд в случае неправильно предсказанного ветвления. Исполнительное ядро работает с повышенной скоростью выполнения операций.

Адреса операндов, выбираемых из памяти, вычисляются блоком формирования адреса (БФА), который реализует интерфейс с кэш-памятью данных L1 ёмкостью 8 Кбайт. В соответствии с заданными в декодированных командах способами адресации формируются 48 адресов для загрузки операндов из памяти в регистр БРЗ и 24 адреса для записи из регистра в память. При этом БФА формирует адреса операндов для команд, которые ещё не поступили на выполнение.

При обращении к памяти БФА одновременно выдаёт адреса двух операндов: один для загрузки операнда в заданный регистр БРЗ, второй - для пересылки результата из БРЗ в память. Таким

образом реализуется процедура предварительного чтения данных для последующей их обработки в исполнительных блоках, которая называется спекулятивной выборкой.

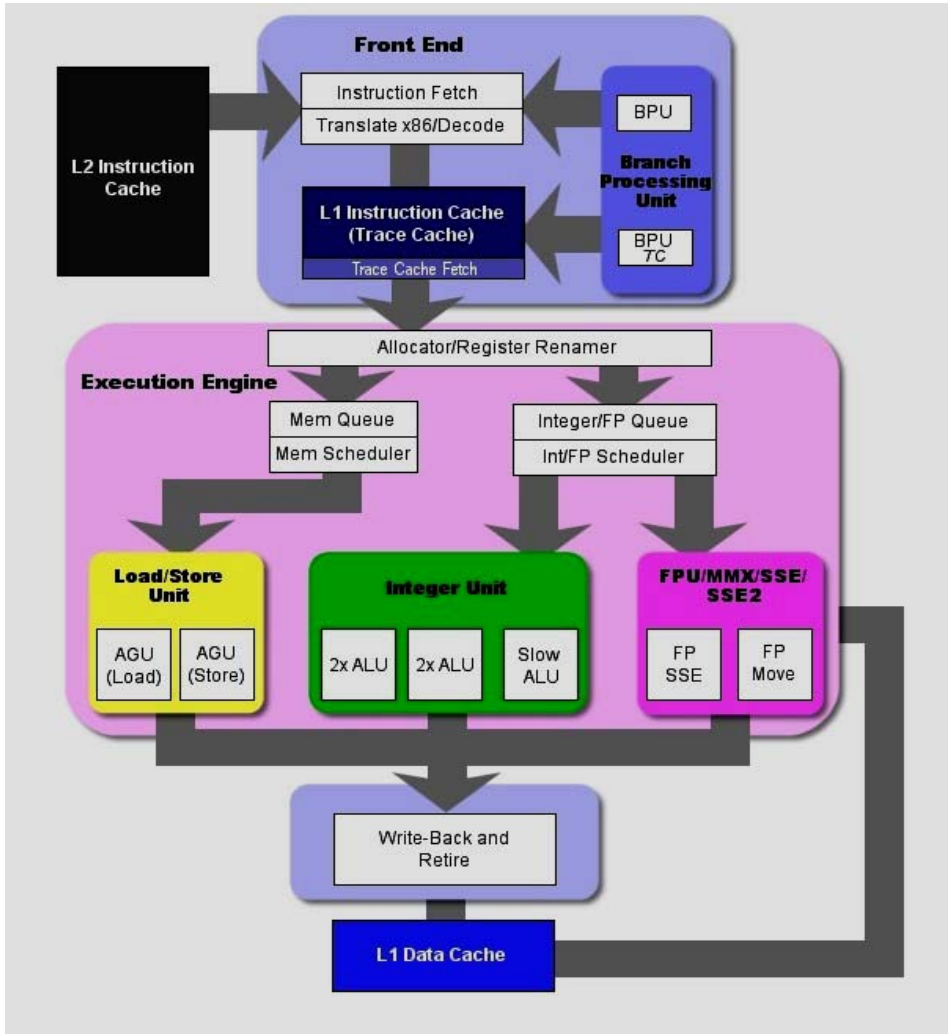


Рис. 7

Аналогичным образом организуется параллельная работа блоков SSE, FPU, MMX, которые используют отдельный набор регистров и блок формирования адресов операций.

Хотя конвейер Pentium 4 является намного более длинным, он выполняет те же функции что и конвейеры других процессоров.

Из-за сложности архитектуры на рис. 7 не изображена каждая из ступеней конвейера. Тем не менее, связанные ступени сгруппированы воедино, чтобы можно было представить всю схему процессора и схему потока команд.

Особое внимание стоит уделить тому, что кэш-память L1 разделена и кэш инструкций находится фактически на препроцессоре. Он называется отслеживающим кэшем (Trace Cache) и является одной из важных инноваций в Pentium 4. Эта кэш-память оказывает сильное влияние и на конвейер, и на основной поток инструкций.

Если рассмотреть процессоры Pentium III или Athlon, то можно отметить, что инструкции поступают в их декодер из кэш-памяти инструкций, в декодере они разбиваются на меньшие части, более единообразные, с которыми проще работать, - микрокоманды. Фактически, эти инструкции применяются при внеочередном выполнении команд, исполнительный модуль выполняет их планирование, исполнение и сброс. Такое разбиение случается всякий раз, когда процессор выполняет инструкцию, поэтому на эту операцию в начале конвейера отводится несколько ступеней (на рис. 8 и 9 эти ступени объединены, на самом же деле выборка инструкций занимает несколько ступеней, транслирование - несколько ступеней, декодирование - несколько, и т.д.).

Если взять фрагмент кода, повторно выполняющийся всего несколько раз по ходу программы, то для него такая потеря нескольких тактов мало что означает. Но для фрагмента кода, где инструкции исполняются тысячи и тысячи раз (например, в цикле в мультимедийном приложении, выполняющем несколько операций над большим файлом), количество повторных трансляций и декодирований может отнимать ощутимые ресурсы. Для того, чтобы избежать таких циклов, процессор Pentium 4 не осуществляет повторного разбиения 80x86-инструкций на микрокоманды при их выполнении.

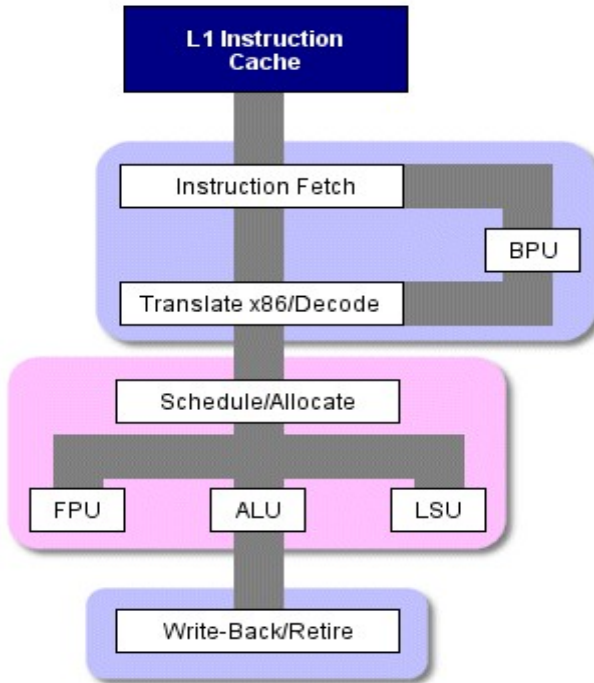


Рис. 8

Кэш инструкций в Pentium 4 принимает транслированные и декодированные микрокоманды, готовые к передаче на внеочередное выполнение, и формирует из них мини-программы, назовем их трейс-последовательностями (traces). Именно эти мини-программы (а не 80x86-код, созданный компилятором) и выполняет Pentium 4 в том случае, если происходит попадание в кэш-память L1 (процент попадания - 90%). До тех пор, пока требуемый код находится в кэш-памяти L1. Схема выполнения представлена на рис. 9.

По мере выполнения препроцессором накопленных трейс-последовательностей, отслеживающий кэш посылает до трех микрокоманд за такт напрямую на внеочередной модуль выполнения, ведь процессору уже не нужно проводить команды через логику трансляции или декодирования. И только в случае промаха кэш-

памяти L1 препроцессор нарушит этот порядок и начнёт выбирать и декодировать инструкции из кэш-памяти L2.

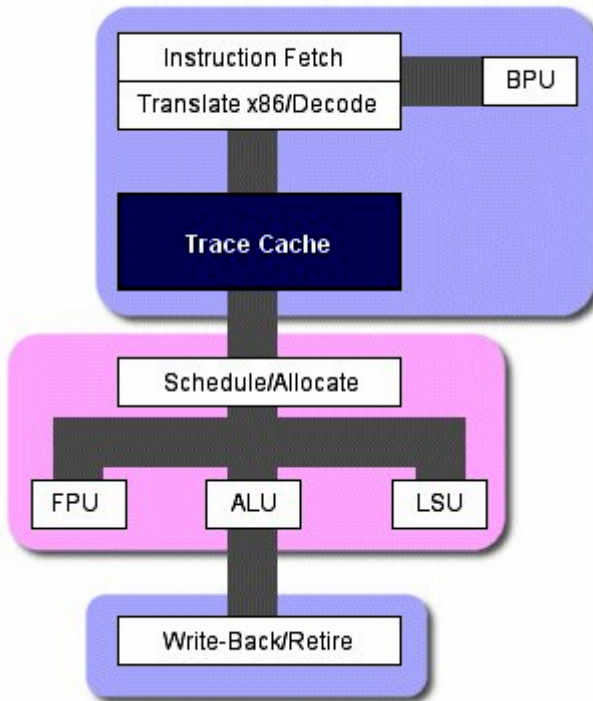


Рис. 9

В этом случае к началу основного конвейера добавляется ещё восемь ступеней. Как видно, кэш с отслеживанием может избавить от довольно большого количества тактов при выполнении программы.

Отслеживающий кэш может работать в двух режимах. Исполнительный режим (execute mode) был только что рассмотрен. Здесь отслеживающий кэш снабжает логику выполнения инструкциями. В этом режиме он обычно и работает. Когда наступает промах кэш-памяти L1, он переходит в так называемый режим построения отслеживаемых сегментов (trace segment build mode) В этом режиме препроцессор выбирает 80x86-инструкции из кэш-памяти L2, транслирует их в микрокоманды, создает отслеживаемые

мый сегмент, который затем перемещается в отслеживающий кэш и далее выполняется.

Из рис. 9 видно, что когда работает отслеживающий кэш, устройство предсказания ветвлений не участвует в работе, равно как не работают и ступени выборки/декодирования инструкций. На самом деле отслеживаемый сегмент - это нечто большее, чем просто фрагмент транслированного и декодированного кода 80x86, выданного компилятором и полученного препроцессором из кэш-памяти L2. В действительности, при создании мини-программы отслеживающий кэш все же использует предсказание ветвлений. Он может добавить в мини-программу (где содержится предназначенный для выполнения код) код, который только предполагается к выполнению при предсказании ветвления. Поэтому если имеется фрагмент x86-кода с ветвлением, отслеживающий кэш построит трейс-последовательность из инструкций до ветвления, включая саму инструкцию ветвления. Затем он продолжит спекулятивно строить мини-программу вдоль предсказанной ветви (рис. 10).

Такое спекулятивное выполнение даёт отслеживающему кэшу два больших преимущества по сравнению с обычным кэшем инструкций. Во-первых, в стандартном процессоре для работы устройства предсказания ветвлений требуется некоторое время. При обработке условной инструкции ветвления устройство ветвления (ВРУ) должно определить, какую из ветвей нужно спекулятивно выполнять, найти адрес кода после ветвления и т.д. Весь этот процесс добавляет, по крайней мере, еще один такт задержки для каждой условной инструкции ветвления. Такая задержка часто не может быть заполнена выполнением другого кода, что приводит к появлению нежелательного пузырька (пустая ячейка в конвейере - pipeline bubble). В случае же использования отслеживающего кэша, код после ветвления уже готов к выполнению сразу же после инструкции ветвления, поэтому показанных задержек не возникает.

Второе преимущество также связано с возможностью хранения спекулятивных ветвей. Когда стандартный кэш инструкций L1 считывает строку кэш-памяти, он прекращает считывание при попадании на инструкцию ветвления, поэтому оставшаяся часть

строки остается пустой. Если инструкция ветвления находится в начале строки кэш-памяти L1, то в считанной строчке будет находиться только одна эта инструкция. При использовании отслеживающего кэша считанные строчки могут содержать как инструкции ветвления, так и спекулятивный код после них. Таким образом, в 6-командных строчках не возникает потерянного места.

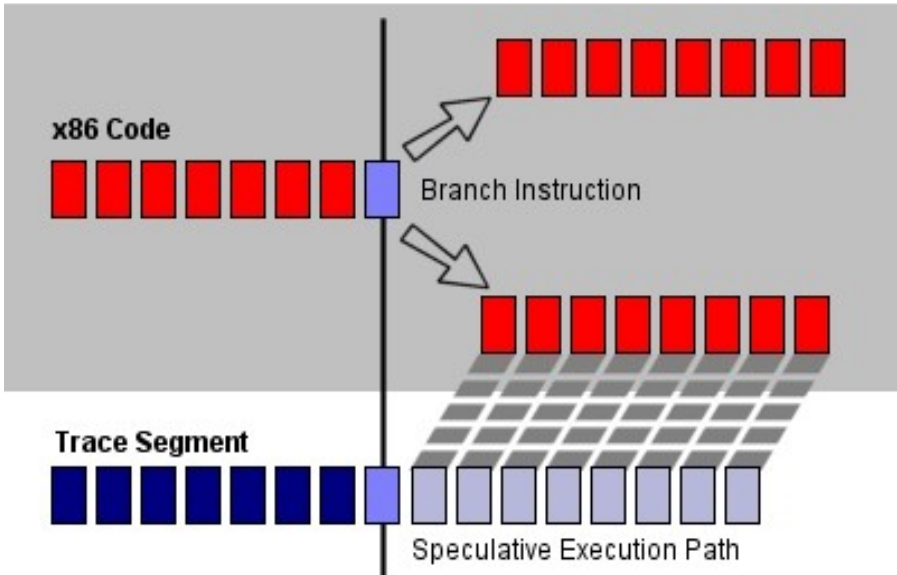


Рис.10.

Кстати, большинство компиляторов сталкиваются именно с описанными двумя проблемами: с задержками в инструкциях ветвления и с неполными строками из кэш-памяти. Как видно, отслеживающий кэш по-своему позволяет решать эти проблемы. Если программы оптимизированы с учетом этих возможностей, то они будут быстрее выполняться.

Ещё один интересный эффект, производимый отслеживающим кэшем на препроцессор Pentium 4 заключается в том, что пропускная способность ступеней транслирования и декодирования 80x86-команд не зависит от пропускной способности ступени диспетчеризации. Если вспомнить процессор AMD K7, то он рас-

ходуется множество транзисторов на усиленный блок декодирования 80x86-макрокоманд, что позволяет за цикл декодировать достаточно много громоздких 80x86-инструкций в микрокоманды для загрузки исполнительного модуля. В случае же с Pentium 4 наличие отслеживающего кэша означает, что большая часть кода забирается из него уже в виде готовых микрокоманд, так что здесь отпадает надобность в трансляторах и декодерах с высокой пропускной способностью.

Процессор начинает декодирование только лишь в случае промаха кэш-памяти L1. Поэтому он разработан таким образом, чтобы декодировать только одну 80x86-инструкцию за такт. Это составляет всего треть от максимальной теоретической пропускной способности декодера Athlon, но отслеживающий кэш в Pentium 4 позволяет ему достичь или даже обойти производительность Athlon (2,5 диспетчеризации за такт).

Стоит обратить внимание и на то, как отслеживающий кэш обращается с очень длинными 80x86-инструкциями из нескольких циклов. Большинство 80x86-инструкций декодируются примерно в две или три микрокоманды. Но встречаются и такие инструкции, которые декодируются в сотни микрокоманд, например, инструкции по строковой обработке. Как и в Athlon, в Pentium 4 существует специальное ПЗУ микрокода, которое обрабатывает эти громоздкие инструкции, что позволяет разгрузить аппаратный декодер для работы только с небольшими, быстрыми инструкциями.

Каждый раз, когда встречается громоздкая инструкция, ПЗУ находит готовую последовательность микрокоманд и выдаёт их дальше в по очереди. Чтобы не засорять отслеживающий кэш этими длинными последовательностями микрокоманд, разработчики поступили следующим образом: как только при создании отслеживаемого сегмента отслеживающий кэш встречает такую большую 80x86-инструкцию, вместо того, чтобы разбивать её на последовательность микрокоманд, он вставляет в отслеживаемый сегмент метку (tag), которая указывает на место в ПЗУ, содержащее последовательность микрокоманд данной инструкции. Позднее, в режиме выполнения, когда отслеживающий кэш будет передавать поток инструкций на ступень выполнения, при попадании на такую метку он временно приостановит работу и на время пе-

редает управление потоком инструкций ПЗУ микрокода. Здесь уже ПЗУ будет выдавать в поток инструкций требуемую последовательность микрокоманд (как определено меткой). После этого, оно возвратит управление обратно, и отслеживающий кэш продолжит передавать инструкции. Исполнительному модулю безразлично, откуда поступает поток инструкций (из отслеживающего кэша или из ПЗУ). Для него все это выглядит как непрерывный поток команд.

Единственным недостатком отслеживающего кэша является его размер: он слишком мал. Точные размеры его неизвестны. Он может содержать до 12 тысяч микрокоманд. Intel уверяет, что это примерно эквивалентно обычному кэшу команд на 16-18 тысяч инструкций. Но так как отслеживающий кэш работает совсем иначе, нежели стандартный кэш инструкций L1, то для того, чтобы оценить, как его размер влияет на производительность всей системы, нельзя обойтись простым сравнением его размера с кэш-памятью другого процессора.

Общая архитектура процессора определяет комплекс средств, предоставляемых пользователю для решения различных задач. Эта архитектура задаёт базовую систему команд процессора и реализуемых способов адресации, набор программно-доступных регистров (регистровая модель), возможные режимы работы процессора и обращения к памяти и внешним устройствам (организация памяти и реализация обмена по системной шине), средства обработки прерываний и исключений.

В процессоре Pentium 4 реализуется архитектура IA-32 (Intel Architecture-32), общая для всех 32-разрядных микропроцессоров Intel, начиная с 80386. Модели Pentium II Xeon и Pentium III Xeon ориентированы на работу в высокопроизводительных мультипроцессорных системах (серверах, рабочих станциях). Для этих же приложений в 2001 году выпущена модификация процессора Pentium 4 с поддержкой мультипроцессорного режима работы (на ядре Foster).

В процессе развития IA-32 производилось расширение возможностей обработки данных, представленных в различных форматах (рис. 11). Процессоры 80386 выполняли обработку только целочисленных операндов. Для обработки чисел с плавающей точ-

кой использовался внешний сопроцессор 80387, подключаемый к микропроцессору. В состав процессоров 80486 и последующих моделей Pentium введён специальный блок FPU (Floating-Point Unit), выполняющий операции над числами с "плавающей точкой". В процессорах Pentium MMX была впервые реализована групповая обработка нескольких целочисленных операндов разрядностью 1, 2, 4 или 8 байт с помощью одной команды. Такая обработка обеспечивается введением дополнительного блока MMX.



Рис. 11

Название блока отражает его направленность на обработку видео- и аудиоданных, когда одновременное выполнение одной операции над несколькими операндами позволяет существенно повысить скорость обработки изображений и звуковых сигналов. Начиная с модели Pentium III, в процессоры вводится блок SSE (Streaming SIMD Extension) для групповой обработки чисел с плавающей точкой.

Таким образом, если первые модели процессоров Pentium выполняли только пооперандную обработку данных по принципу

"одна команда - одни данные" (SISD), то, начиная с процессора Pentium MMX, реализуется также их групповая обработка по принципу "одна команда - много данных" (SIMD).

Соответственно, расширяется и набор регистров процессора, используемых для промежуточного хранения данных (рис. 12). Кроме 32-разрядных регистров для хранения целочисленных операндов, процессоры Pentium содержат 80-разрядные регистры, которые обслуживают блоки FPU и MMX. При работе FPU регистры ST0-ST7 образуют кольцевой стек, в котором хранятся числа с плавающей точкой, представленные в формате с расширенной точностью (80 разрядов). При реализации MMX-операций они используются как 64-разрядные регистры MM0-MM7, где могут храниться несколько операндов (8 8-разрядных, 4 16-разрядных, 2 32-разрядных или один 64-разрядный), над которыми одновременно выполняется поступающая в процессор команда (арифметическая, логическая, сдвиг и т.д.).

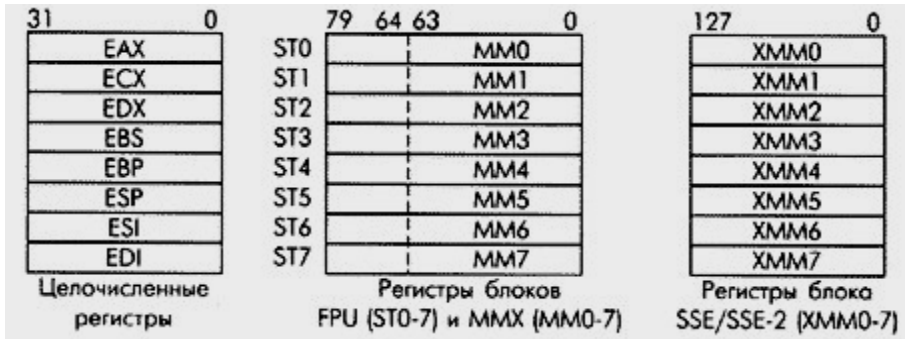


Рис.12

Блок SSE2, введённый в состав процессора Pentium 4, значительно расширяет возможности обработки нескольких операндов по принципу SIMD, по сравнению с блоком SSE в модели Pentium III. Этот блок реализует 144 новые команды, обеспечивающих одновременное выполнение операций над несколькими операндами, которые располагаются в памяти и в 128-разрядных регистрах XMM0-XMM7. В регистрах могут храниться и одновре-

менно обрабатываться 2 числа с плавающей точкой в формате двойной точности (64 разряда) или 4 числа в формате одинарной точности (32 разряда). Этот блок может также одновременно обрабатывать целочисленные операнды: 16 8-разрядных, 8 16-разрядных, 4 32-разрядных или 2 64-разрядных. В результате производительность процессора Pentium 4 при выполнении таких операций оказывается вдвое выше, чем Pentium III.

Операции SSE2 позволяют существенно повысить эффективность процессора при реализации трехмерной графики и интернет-приложений, обеспечении сжатия и кодирования аудио- и видео-данных. Что касается базового набора команд и используемых способов адресации операндов, то они практически полностью совпадают с набором команд и способов адресации в предыдущих моделях Pentium. Процессор обеспечивает реальный и защищенный режимы работы, реализует 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 семейства Pentium.

Pentium 4 является первым IA-32 (32-bit Intel Architecture) процессором, использующим не Р6 архитектуру. Сегодня эта архитектура получает название с использованием терминологии - Intel NetBurst.

Первой особенностью NetBurst архитектуры является то, что Intel называет гиперконвейерной технологией, что является несколько причудливым термином для 20-ти шагового конвейера Pentium 4. Эти 20 шагов или стадий – вдвое длиннее Р6 конвейера, которым оснащен Pentium III и в четыре раза длиннее, чем Р5 конвейер. Как известно, более длинный конвейер имеет свои “за” и “против”.

20-ти шаговый конвейер Pentium 4, позволяет ему работать на более высокой тактовой частоте. По этой причине Pentium 4 будет дебютировать на скорости 1.4 ГГц и выше. Но 20-ти шаговый конвейер Pentium 4 приводит к уменьшению значения IPC (инструкций за такт).

Имеется множество путей восполнения низкого IPC. Один из них, наиболее очевидный, заключается в простом увеличении тактовой частоты, что Intel и сделала. Нет сомнения, что на любом современном эталонном тесте 1ГГц Pentium III по сравнению с гипотетическим 1ГГц Pentium 4 показал бы значительно больший результат, потому что выполняет больше инструкций за такт, чем Pentium 4.

В современных процессорах предусмотрены средства увеличения эффективности конвейеров за счет предсказания хода программы. Когда процессор правильно предсказывает следующую команду, все идет согласно плану, но когда предсказание сделано неверно, цикл обработки должен начаться с начала. Из-за этого процессор с 10 шаговым конвейером имеет несколько меньший штраф за неправильный переход, чем процессор с 20 шаговым конвейером.

Для уменьшения недостатков более длинного конвейера, архитектура NetBurst имеет несколько особенностей.

Как уже указывалось ранее, АЛУ Pentium 4 работает на удвоенной тактовой частоте. Это означает, что АЛУ 1.4ГГц Pentium 4 работает на 2.8ГГц, а у 1.5ГГц Pentium 4 на 3.0ГГц. Считается, что это дает Pentium 4 явное преимущество в производительности в целочисленных операциях. Однако практика показала, что основной причиной удвоенной частоты АЛУ является восполнение более низкий IPC NetBurst архитектуры.

Другая особенность связана со снижением влияния более длинного конвейера заключается в том, что Intel называет Execution Trace Cache. Декодер любого 80x86 процессора (модуль, который берет выбранные инструкции и декодирует их в форму, понятную вычислительным модулям) является одним из самых медленных модулей. Execution Trace Cache действует как посредник между стадией декодирования и первой стадией выполнения. Trace cache по существу кэширует декодированные micro-ops (инструкции после того, как они были выбраны и декодированы, т.е. полностью готовы к выполнению) так, чтобы вместо прохождения процесса выборки и декодирования при выполнении новой команды Pentium 4 мог обратиться к trace cache, получить декодированные micro-ops и начинать выполнение.

Это помогает уменьшать штраф, связанный с неправильно предсказанным переходом в длинном конвейере Pentium 4. Другой особенностью trace cache является, то, что он кэширует micro-ops в предсказанном пути выполнения, означая, что, если Pentium 4 выбрал 3 инструкции из trace cache, то они уже представлены в порядке выполнения. Это добавляет некоторый потенциал для неправильного предсказания пути выполнения кэшируемых micro-ops, однако Intel уверена, что это будет компенсироваться новыми алгоритмами предсказания, используемых в Pentium 4.

Intel отказалась от обычного метода определения размера кэш-памяти, по крайней мере для Execution Trace Cache. Вместо этого заявлено, что trace cache может кэшировать приблизительно 12K micro-ops. В дополнение к Execution Trace Cache, Pentium 4 имеет 8KB L1 Data Cache. Очевидно, что это меньше 16KB L1 Data Cache Pentium III. Такой размер кэш-памяти был сделан для достижения лучшего отношения цены и производительности для Pentium 4.

Pentium 4 также имеет 256KB L2, работающий на основной тактовой частоте процессора. Этот кэш имеет большую ширину полосы частот, чем текущий 256KB L2 в Pentium III. Основной причиной является работа на большей тактовой частоте, а также передача данных на каждом такте.

В терминах полосы частот, доступной L2, гипотетический Pentium III работающий на тактовой частоте 1.5ГГц имел бы скорость передачи 24GB/s. Pentium 4 на той же тактовой частоте имеет 48GB/s. L1 в Pentium 4 (включая Execution Trace Cache) дублируется в L2.

5.2.2. AMD Athlon

Микроархитектура Athlon представлена на рис.13. С целью увеличения пропускной способности декодеров, которые перекодируют 80x86-команды во внутренние макрооперации, в блок кэш-памяти команд первого уровня добавлена специальная кэш-память предварительного декодирования. Эти макрооперации представляют собой по сути RISC-команды, которые, собственно, и испол-

няются. Подобная схема перекодирования из 80x86 в RISC используется и в Pentium III, и в предыдущих процессорах AMD, и в некоторых других 80x86-совместимых процессорах.

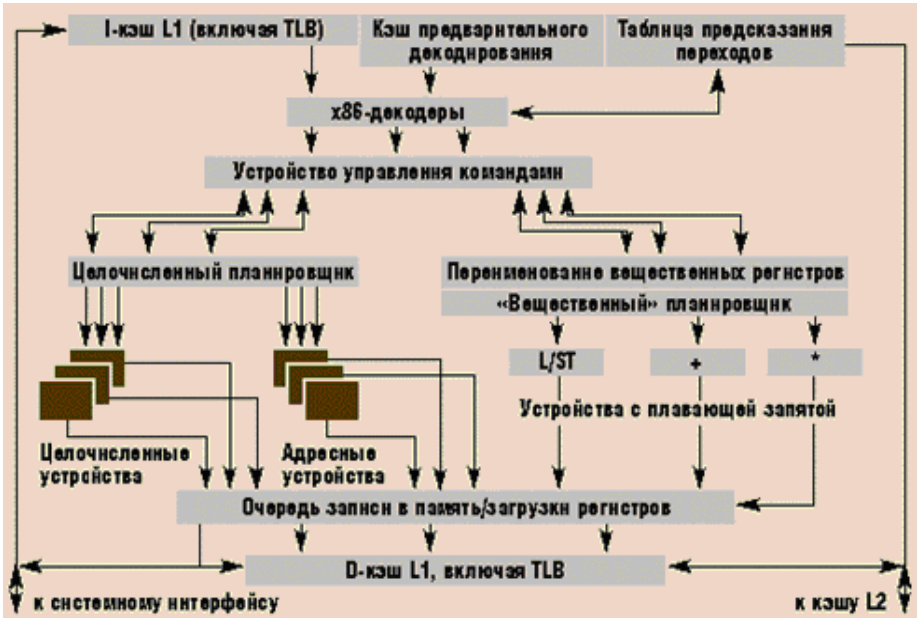


Рис.13

В Athlon таких декодеров три, а число выдаваемых на выполнение за такт команд больше, чем в Pentium III. Команды в декодеры попадают из кэш команд первого уровня. Он является двухканальным, а его емкость составляет 64 Кбайт, что в четыре раза выше, чем в Pentium III. Кроме того, блок кэш команд в AMD содержит два уровня буферов быстрой переадресации TLB: первого уровня — на 24 строки, а второго — на 256.

Емкость кэш данных первого уровня в Athlon также в четыре раза больше, чем в Pentium III, и составляет 64 Кбайт. Он является двухканальным и включает двухуровневый блок TLB. Кэш данных имеет восемь банков, что позволяет одновременно загружать в регистры или писать из них в кэш до двух 64-разрядных величин. Команды из кэш команд поступают в декодеры, а оттуда —

в блок управления командами, емкость которого 72 строки.

Athlon является суперскалярным микропроцессором с внеочередным спекулятивным выполнением команд. Большая емкость очереди команд позволяет Athlon эффективно использовать свои ресурсы: число функциональных исполнительных устройств в нем больше, чем в Pentium III.

Совокупность вышеуказанных факторов уже объясняет, почему производительность Athlon выше, чем в Pentium III. В Athlon имеется три целочисленных устройства и три устройства с плавающей запятой. Кроме того, Athlon содержит три адресных устройства. Все исполнительные устройства способны работать во внеочередном режиме. Для этого в арифметических устройствах имеются «планировщики», содержащие очереди команд емкостью 18 (для целочисленных устройств) и 36 (для вещественных устройств) строк соответственно.

Следует отметить, что в составе МП имеются следующие устройства с плавающей запятой: сумматор, умножитель и модуль загрузки регистров/записи в память. Они могут работать параллельно, поэтому производительность микропроцессора (в MFLOPS) в два раза выше, чем его тактовая частота (1,3 GFLOPS при 650 МГц), и в два раза выше, чем у Pentium III при той же частоте. Кроме того, умножитель Athlon - это настоящий конвейер. Физических регистров с плавающей запятой в Athlon - 88, что позволяет использовать технологию переименования регистров.

Athlon не только суперскалярный, но еще так называемый и суперконвейерный микропроцессор. С одной стороны, большое число ступеней конвейеров (10 — в целочисленном и 15 — в вещественном конвейере) позволяет легче поднимать тактовую частоту. С другой стороны, это вызывает проблему заполнения конвейеров: если они не заполняются, производительность падает. Наиболее «опасными» будут при этом программы нерегулярного характера с большим числом условных переходов, которые трудно динамически предсказывать. Кстати, в Athlon блок динамического предсказания переходов включает таблицу предыстории на 2048 строк. Такой большой объем позволяет добиться очень высокого качества предсказания переходов.

В Athlon расширена система команд 3D-Now!. Их теперь

45, из них 24 - новых, в том числе: 12 команд целочисленной математики для обработки видео и распознавания речи; 7 команд пересылки данных, ориентированных на программы, подобные Internet-приложениям, работающим с графическими данными; 5 новых команд для цифровой обработки сигналов

Кроме рассмотренных выше блоков микропроцессора, Athlon имеет встроенное управление внешней кэш-памятью второго уровня и сопряжение с системной шиной. Интеграция в микропроцессор функций управления внешней кэш-памятью позволяет Athlon иметь L2, расположенный на выделенной шине с программируемой частотой. Поддерживается совместимость с промышленными стандартами SRAM, в том числе DDR и SDR. Кроме того, этот блок содержит память тегов для L2 наиболее популярного размера 512 Кбайт, при этом емкость L2 может составлять до 8 Мбайт.

Системный интерфейс Athlon обеспечивает соединения «точка-точка», то есть фактически мы имеем дело с коммутатором, а не с общей системной шиной, как у Pentium III. Важным преимуществом коммутаторов является то, что в отличие от системной шины они не имеют конфликтов и обеспечивают гарантированный уровень пропускной способности. Это особенно важно для многопроцессорных SMP-систем.

Системный интерфейс шириной 8 байт может работать на частотах от 200 до 400 МГц, что обеспечивает гораздо более высокую пропускную способность, чем у шины Pentium III. Однако, по некоторым данным, на тестах пропускной способности оперативной памяти (STREAM) Athlon лишь незначительно опережает Pentium III. Важным преимуществом системного интерфейса Athlon является расщепленная обработка транзакций (до 24 на процессор против 4 в Pentium III). При этом пакетный протокол может передавать блоки в 64 байт против 32 байт у Pentium III.

Максимальная поддерживаемая емкость оперативной памяти составляет у Athlon 7 Тбайт против 64 Гбайт в Pentium III; впрочем, это отличие вряд ли имеет сегодня практическое значение. Учитывая высокую пропускную способность системного интерфейса, он явно проектировался в расчете на использование с Athlon технологии RAMBUS. Увеличилась и надежность: теперь

как шина внешней кэш-памяти, так и системный интерфейс используют ECC-коды.

Athlon имеет площадь 128 кв. мм и производится по 0,18-микронной технологии с шестислойной металлизацией. Он использует разъем типа Slot A, механически совместимый со Slot 1 и близкий к применяемому в Alpha EV6.

5.2.3. MC88110 компании Motorola

Процессор 88110 относится к разряду суперскалярных RISC-процессоров. Основные особенности этого процессора связаны с использованием принципов суперскалярной обработки, двух восьмипортовых регистровых файлов, десяти независимых исполнительных устройств, больших по объему внутренних кэш-памятей и широких магистралей данных.

На рис. 14 представлена блок-схема процессора, содержащего 1.3 миллиона вентиляей. Центральной частью этой архитектуры является шина операндов (в реализации это шесть 80-битовых шин), соединяющая регистровые файлы и исполнительные устройства.

Процессор имеет 10 исполнительных устройств, которые работают одновременно и независимо, и два регистровых файла. Файл регистров общего назначения имеет 32-битовую организацию. Расширенные регистры плавающей точки имеют 80-битовую организацию. Эти регистровые файлы снабжены шестью портами чтения и двумя портами записи каждый.

Внешняя шина процессора имеет отдельные линии данных (64 бит) и адреса (32 бит), что позволяет реализовать быстрые групповые операции перезагрузки внутренней кэш-памяти. Внешняя шина имеет также специальные сигналы управления, обеспечивающие аппаратную поддержку когерентности кэш-памяти в мультипроцессорных конфигурациях.

В процессоре имеются две двухканальные множественно-ассоциативные кэш-памяти емкостью по 8 Кбайт (для команд и для данных). Они имеют физическую адресацию.

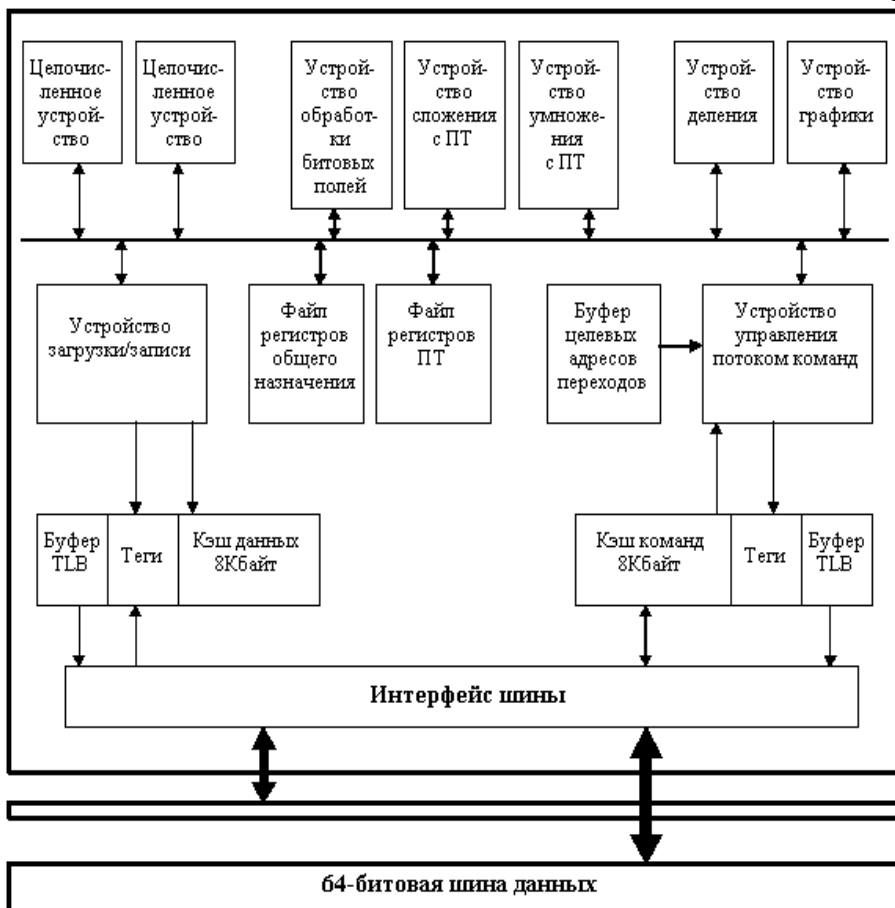


Рис. 14

Все операции по перезагрузке кэш-памяти выполняются в режиме групповой пересылки данных, при этом первым пересылается требуемое слово. Когерентность кэш данных обеспечивается аппаратным протоколом наблюдения за шиной с четырьмя состояниями (MESI). Для увеличения производительности в кэш памяти данных применяется стратегия задержанного обратного копирования.

Суперскалярная архитектура процессора базируются на реализации возможности завершения команд не в порядке их поступления для выполнения, что позволяет существенно увеличить производительность, однако приводит к проблемам организации точного прерывания. Эта проблема решается в процессоре 88110 с помощью так называемого буфера истории, который хранит старые значения регистров при выполнении и завершении операций не в предписанном программой порядке, и позволяет аппаратно восстановить необходимое состояние в случае прерывания.

В процессоре предусмотрено несколько способов ускорения обработки условных переходов. Один из них, предсказание направления перехода, позволяет компилятору сообщить процессору предпочтительное направление перехода.

Для выполняемых переходов используется буфер целевых адресов перехода емкостью 32 строки, позволяющий быстро выбрать две команды по целевому адресу перехода. Механизм предсказания направления переходов позволяет одновременно выполнять эти команды и оценивать условие перехода. Для предсказанного направления перехода разрешено спекулятивное (условное) выполнение команд. Если направление перехода предсказано неверно, исходное состояние процессора восстанавливается с помощью буфера истории. Выполнение программы в этом случае будет продолжено с "правильной" команды.

В каждом такте процессор может выдавать на выполнение две команды. В большинстве случаев выдача команд осуществляется в порядке, предписанном программой. Команды записи и условных переходов могут посылаются на буферные станции резервирования, из которых они в дальнейшем будут выданы на выполнение. Команды загрузки могут накапливаться в очереди. Таким образом эти команды не блокируют выдачу второй команды из пары.

Большое количество исполнительных устройств позволяет осуществлять одновременную выдачу двух команд во многих ситуациях: 2 целочисленные команды, 2 команды с плавающей точкой, 2 графические команды или любая комбинация перечисленных команд.

В устройстве загрузки/записи реализован буфер загрузки FIFO на четыре строки и три станции резервирования операций записи, что позволяет иметь в каждый момент времени до 4 отложенных команд загрузки и до трех команд записи. Выполнение этих команд внутри устройства может переупорядочиваться для обеспечения большей эффективности.

При построении многопроцессорной системы все процессоры и основная память размещаются на одной плате. Для обеспечения хорошей производительности системы каждый процессор в такой конфигурации снабжается кэш-памятью второго уровня емкостью 256 Кбайт. Протокол поддержания когерентного состояния кэш-памяти (протокол наблюдения) базируется на методике записи с аннулированием, гарантирующей размещение модифицированной копии строки кэш памяти только в одной из кэш системы. Протокол позволяет нескольким процессорам иметь одну и ту же копию строки кэш памяти. При этом, если один из процессоров выполняет запись в память (общую строку кэш памяти), другие процессоры уведомляются о том, что их копии являются недействительными и должны быть аннулированы.

5.3. Микропроцессоры с масштабируемой архитектурой

5.3.1. SuperSPARC

Масштабируемая процессорная архитектура компании Sun Microsystems (SPARC - Scalable Processor Architecture) является наиболее широко распространенной RISC-архитектурой, отражающей доминирующее положение компании на рынке UNIX-рабочих станций и серверов. Процессоры с архитектурой SPARC лицензированы и изготавливаются по спецификациям Sun несколькими производителями, среди которых следует отметить компании Texas Instruments, Fujitsu, LSI Logic, Bipolar International Technology, Philips и Cypress Semiconductor. Эти компании осуществляют поставки процессоров SPARC не только самой Sun Microsystems, но и другим известным производителям вычисли-

тельных систем, например, Solbourne, Toshiba, Matsushita, Tatung и Cray Research.

Первоначально архитектура SPARC была разработана с целью упрощения реализации 32-битового процессора. В последствии по мере улучшения технологии изготовления интегральных схем она постепенно развивалась и в настоящее время имеется 64-битовая версия этой архитектуры.

В отличие от большинства RISC архитектур, SPARC использует регистровые окна, которые обеспечивают удобный механизм передачи параметров между программами и возврата результатов. Архитектура SPARC была первой коммерческой разработкой, реализующей механизмы отложенных переходов и аннулирования команд. Это давало компилятору большую свободу заполнения времени выполнения команд перехода командой, которая выполняется в случае выполнения условий перехода и игнорируется в случае, если условие перехода не выполняется.

Первый процессор SPARC был изготовлен компанией Fujitsu на основе вентильной матрицы, работающей на частоте 16.67 МГц. На основе этого процессора была разработана первая рабочая станция Sun-4 с производительностью 10 MIPS, объявленная осенью 1987 года (до этого времени компания Sun использовала в своих изделиях микропроцессоры Motorola 680X0). В марте 1988 года Fujitsu увеличила тактовую частоту до 25 МГц создав процессор с производительностью 15 MIPS.

Позднее компания Sun умело использовала конкуренцию среди компаний-поставщиков интегральных схем (LSI Logic, Cypress и Texas Instruments), выбирая наиболее удачные разработки для реализации своих изделий SPARCstation 1, SPARCstation 1+, SPARCstation IPC, SPARCstation ELC, SPARCstation IPX, SPARCstation 2 и серверов серий 4XX и 6XX. Тактовая частота процессоров SPARC была повышена до 40 МГц, а производительность - до 28 MIPS.

Дальнейшее увеличение производительности процессоров с архитектурой SPARC было достигнуто за счет реализации в кристаллах принципов суперскалярной обработки компаниями Texas Instruments и Cypress. Процессор SuperSPARC компании Texas Instruments стал основой серии рабочих станций и серверов

SPARCstation/SPARCserver 10 и SPARCstation/SPARCserver 20.

Имеется несколько версий этого процессора, позволяющего в зависимости от смеси команд обрабатывать до трех команд за один машинный такт, отличающихся тактовой частотой. Процессор SuperSPARC (рис. 15) имеет сбалансированную производительность на операциях с фиксированной и плавающей точкой. Он имеет внутреннюю кэш-память емкостью 36 Кб (20 Кб - кэш команд и 16 Кб - кэш данных), отдельные конвейеры целочисленной и вещественной арифметики и при тактовой частоте 75 МГц обеспечивает производительность около 205 MIPS. Процессор SuperSPARC применяется также в серверах SPARCserver 1000 и SPARCcenter 2000 компании Sun.

Конструктивно кристалл монтируется на взаимозаменяемых процессорных модулях трех типов, отличающихся наличием и объемом кэш-памяти второго уровня и тактовой частотой. Модуль M-bus SuperSPARC, используемый в модели 50, содержит 50-МГц SuperSPARC процессор с внутренней кэш-памятью емкостью 36 Кб (20 Кб кэш команд и 16 Кб кэш данных). Модули M-bus SuperSPARC в моделях 51, 61 и 71 содержат по одному SuperSPARC процессору, работающему на частоте 50, 60 и 75 МГц соответственно, одному кристаллу кэш-контроллера (так называемому SuperCache), а также внешнюю кэш-память емкостью 1 Мб.

Модули M-bus в моделях 502, 612, 712 и 514 содержат два SuperSPARC процессора и два кэш-контроллера каждый, а последние три модели и по одному 1 Мб внешней кэш-памяти на каждый процессор. Использование кэш-памяти позволяет модулям CPU работать с тактовой частотой, отличной от тактовой частоты материнской платы. Пользователи всех моделей поэтому могут улучшить производительность своих систем заменой существующих модулей CPU, вместо того, чтобы производить upgrade всей материнской платы.

Компания Texas Instruments разработала также 50 МГц процессор MicroSPARC с встроенной кэш-памятью емкостью 6 Кб. Он ранее широко использовался в дешевых моделях рабочих станций SPARCclassic и SPARCstation LX, а в настоящее время применяется лишь в X-терминалах.

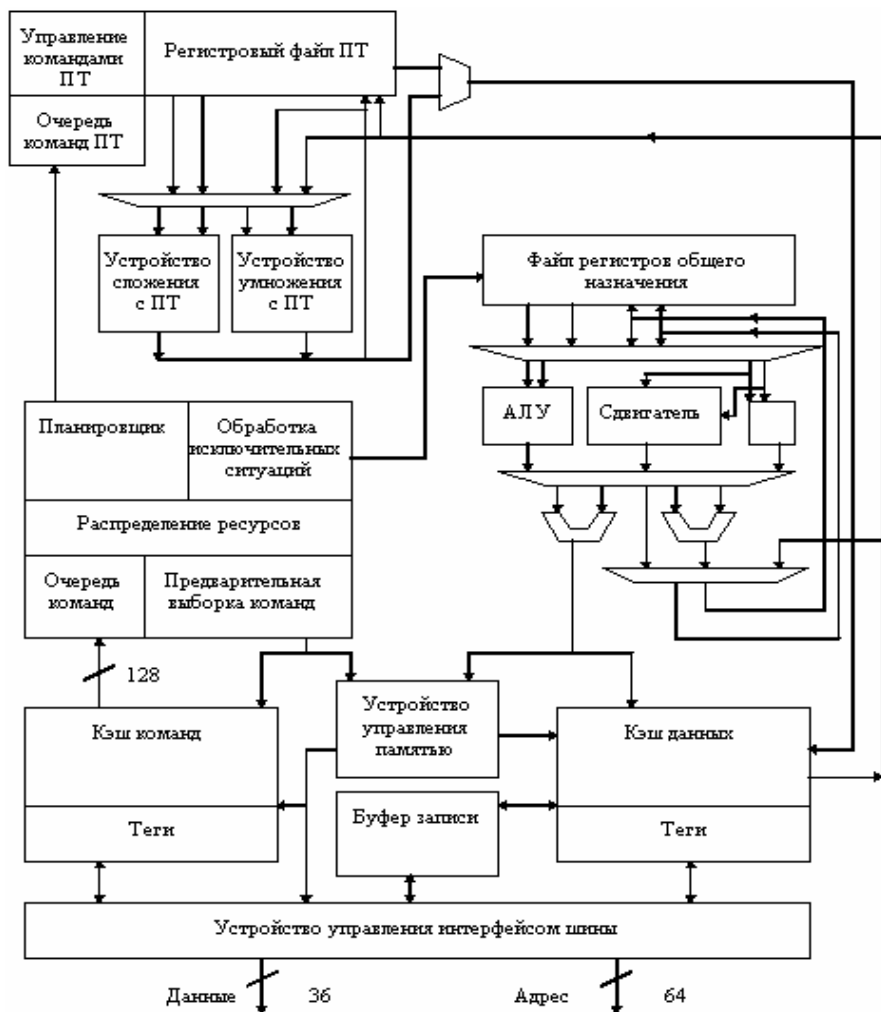


Рис. 15

Sun совместно с Fujitsu создали также новую версию кристалла MicroSPARC II с встроенной кэш-памятью емкостью 24 Кб. На его основе построены рабочие станции и серверы SPARCstation/SPARCserver 4 и SPARCstation/SPARCserver 5, работающие на частоте 70, 85 и 110 МГц.

Хотя архитектура SPARC остается доминирующей на рынке процессоров RISC, особенно в секторе рабочих станций, повышение тактовой частоты процессоров происходило более медленными темпами по сравнению с повышением тактовой частоты других архитектур процессоров. Чтобы ликвидировать это отставание, а также в ответ на появление на рынке 64-битовых процессоров компания, Sun проводила в жизнь программу модернизации. В соответствии с этой программой Sun довела тактовую частоту процессоров MicroSPARC до 100 МГц в 1994 году (процессор MicroSPARC II с тактовой частотой 70, 85 и 110 МГц уже используется в рабочих станциях и серверах SPARCstation 5) и до 125 МГц (процессор MicroSPARC III). В конце 1994 - начале 1995 года на рынке появились микропроцессоры hyperSPARC и однопроцессорные и двухпроцессорные рабочие станции с тактовой частотой процессора 100 и 125 МГц. К середине 1995 года тактовая частота процессоров SuperSPARC доведена до 90 МГц (60 и 75 МГц версии этого процессора в настоящее время применяются в рабочих станциях и серверах SPARCstation 20, SPARCserver 1000 и SPARCcenter 2000 компании Sun и 64-процессорном сервере компании Cray Research). Во второй половине 1995 года появились 64-битовые процессоры UltraSPARC I с тактовой частотой от 167 МГц, в конце 1995 - начале 1996 года - процессоры UltraSPARC II с тактовой частотой от 200 до 275 МГц, а в 1997/1998 годах - процессоры UltraSPARC III с частотой 500 МГц.

5.3.2. MicroSPARC-II

Эффективная с точки зрения стоимости конструкция не может полагаться только на увеличение тактовой частоты. Экономические соображения заставляют принимать решения, основой которых является массовая технология. Системы microSPARC обеспечивают высокую производительность при умеренной тактовой частоте путем оптимизации среднего количества команд, выполняемых за один такт. Это ставит вопросы эффективного управления конвейером и иерархией памяти. Среднее время обращения к памяти должно сокращаться, либо должно возрасти среднее количество команд, выдаваемых для выполнения в каждом такте,

увеличивая производительность на основе компромиссов в конструкции процессора.

MicroSPARC-II (рис. 16) является представителем семейства микропроцессоров SPARC. Основное его назначение - однопроцессорные низкостоимостные системы. Он представляет собой высокоинтегрированную микросхему, содержащую целочисленное устройство, устройство управления памятью, устройство плавающей точки, отдельную кэш-память команд и данных, контроллер управления микросхемами динамической памяти и контроллер шины SBus.

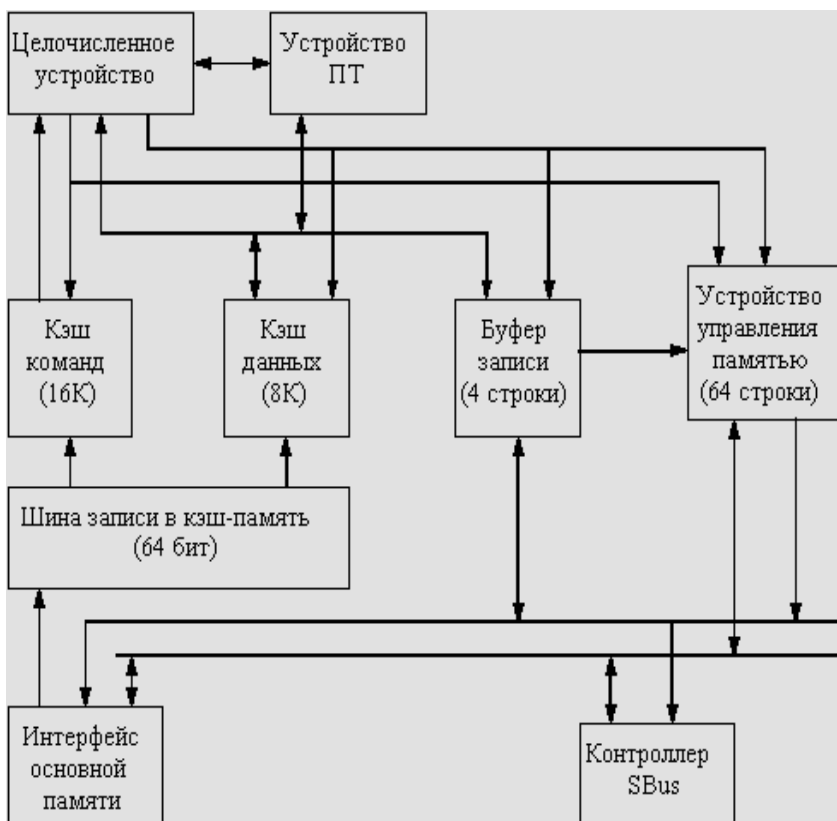


Рис. 16

Основными свойствами целочисленного устройства microSPARC-II являются:

- пятиступенчатый конвейер команд;
- предварительная обработка команд переходов;
- поддержка потокового режима работы кэш-памяти команд и данных;
- регистровый файл емкостью 136 регистров (8 регистровых окон);
- интерфейс с устройством плавающей точки;
- предварительная выборка команд с очередью на четыре команды.

Целочисленное устройство использует пятиступенчатый конвейер команд с одновременным запуском до двух команд. Устройство плавающей точки обеспечивает выполнение операций в соответствии со стандартом IEEE 754.

Устройство управления памятью выполняет четыре основных функции. Во-первых, оно обеспечивает формирование и преобразование виртуального адреса в физический. Эта функция реализуется с помощью ассоциативного буфера TLB. Кроме того, устройство управления памятью реализует механизмы защиты памяти. И, наконец, оно выполняет арбитраж обращений к памяти со стороны ввода/вывода, кэша данных, кэша команд и TLB.

Процессор microSPARC II имеет 64-битовую шину данных для связи с памятью и поддерживает оперативную память емкостью до 256 Мбайт. В процессоре интегрирован контроллер шины SBus, обеспечивающий эффективную с точки зрения стоимости реализацию ввода/вывода.

6. Особенности архитектуры 64 – разрядных МП

6.1. Itanium 2 Intel

Процессор разрабатывался с нуля, причем, параллельно сразу в двух версиях: инженерами Intel и Hewlett-Packard. Впрочем, в основе обоих чипов лежали, естественно, одни и те же идеи, поскольку создавались они все же совместно, и должны были оба стать родоначальниками одного и того же семейства. Цементи-

рующим составом были, естественно, единая идеология, пришедшая на смену CISC - EPIC (Explicitly Parallel Instruction Computing), и новая архитектура - IA-64, включающая в себя набор инструкций, описание регистров, и прочие подобные вещи. Впрочем, архитектура как раз - вещь изменчивая, достаточно вспомнить как отличаются между собой такие CISC процессоры, как 8086 и i80486, оба созданные на базе 80x86.

Точно так же и с Merced и McKinley, Itanium и Itanium 2 - оба построены на базе одной идеологии, но в разных разновидностях архитектуры. В свое время та же история, в общем то, была и с Pentium и Pentium Pro. Впрочем, общие черты были и у тех, есть и у этих, за это "отвечает" EPIC. В первую очередь речь идет о полноценной масштабной суперскалярности, то есть, способности выполнять одновременно несколько инструкций. Для чего, естественно, процессор содержит исполнительные модули - для операций с целыми числами, с числами с плавающей запятой, и т.д.

В отличие от Pentium и его последователей, разбирающихся в коде самостоятельно, EPIC-процессоры сильно полагаются на компилятор, который должен сам проанализировать код на предмет нахождения оптимальных мест для распараллеливания его выполнения, и снабдить процессор этой информацией. Поэтому и используется «explicitly» - процессор не должен сам пытаться понять, что можно выполнять параллельно, а что нет, и т.д. - все это ему уже заранее объяснит компилятор. Плюс, мощные механизмы по предсказанию переходов, предварительному выполнению кусков кода, предварительной загрузке данных, и тому подобные вещи - загрузка исполнительных блоков должна быть распределена максимально равномерно.

Кардинально решен вопрос с регистрами, количество которых увеличено в несколько раз: у Itanium их количество составляет 128 общего назначения (рис.17), 128 - для хранения чисел с плавающей запятой, 8 регистров переходов, и 64, отвечающих за работу механизмов предсказания. Здесь все очевидно - такого количества регистров, да еще реально 64-битных, хватит для хранения любых требуемых чисел для любого разумного количество исполнительных модулей. У Itanium, первого представителя семейства, таких регистров всего пять - два целочисленных, два для операций

с памятью и четыре - для операций с плавающей точкой. Физическая память адресуется 44-бит числами, что на самом деле ограничивает ее объем "всего лишь" 17.6 Терабайт, блоки для операций с плавающей точкой работают с числами в 82-бит представлении.

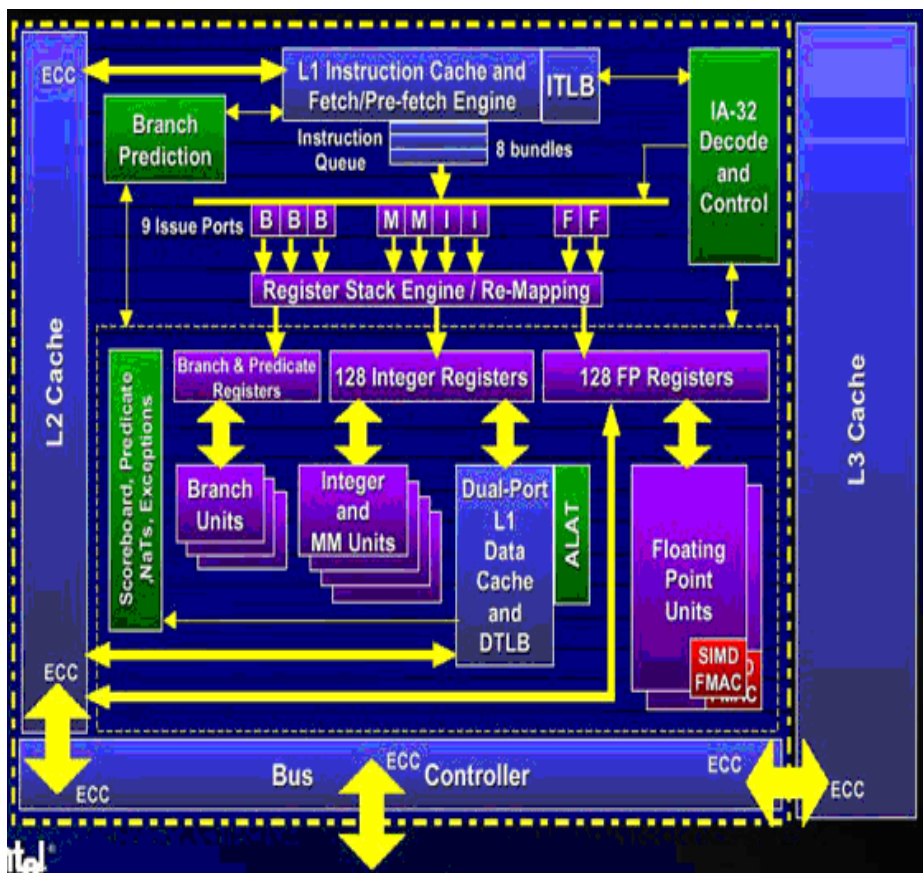


Рис.17

От идеи реализовать 32-бит 80x86 ядро в аппаратном виде Intel отказался, сочтя это слишком неэффективным использованием площади кристалла. Так что для того, чтобы получить возможность исполнения Itanium 80x86 кода, была создана система трансляции, которая на лету преобразует 80x86 код в IA-64.

Очевидно, что при прочих равных, производительность подобного решения будет ниже, чем чистого x86, работающего на той же частоте. Впрочем, никто и не ждал от Itanium скоростного исполнения x86 программ - поддержка этой архитектуры относится скорее к издержкам переходного периода. Тем не менее, факт остается фактом: это семейство для решения 32-бит задач не приспособлено. Впрочем, вряд ли кто-то будет использовать Itanium для подобных целей при наличии полноценного 64-битного ПО..

Вдобавок, сам по себе Itanium был в значительной степени пилотным проектом, как и Pentium Pro, так что процессор вообще стоит рассматривать больше как демонстрацию возможностей архитектуры. Характерный штрих - чипсет для Itanium, 460GX, поддерживает в качестве памяти всего лишь PC100 SDRAM, это кое-что говорит о скорости, с которой способен переваривать данные процессор. С другой стороны, однако, в какой-то мере не слишком быстрый интерфейс с оперативной памятью компенсируется очень большой кэш-памятью L3 - 2 или 4 Мбайт, работающей на полной частоте процессора (733 или 800 МГц) с пропускной способностью до 12.8 Гбайт/с.

Еще одной задачей Itanium было решить вопрос с компиляторами - ведь EPIC-процессоры, как уже упоминалось, очень сильно от них зависят. В отличие от компиляторов для 80x86 процессоров, которые на их производительность почти не влияли, здесь компиляторы являются полноправными партнерами процессора - ведь они снабжают его крайне необходимой для работы информацией, и от того, насколько качественной она будет, будет зависеть скорость исполнения этой программы процессором.



Itanium 2 является уже куда более коммерчески интересным продуктом. Созданный Hewlett-Packard, набившей руку на создании 64-бит процессоров серии PA-

RISC, чип получился куда более совершенным. С несколько меньшим количеством L3 (1.5 или 3 Мбайт) и несколько более

высокой частотой, 900 МГц или 1 ГГц, он обеспечивает в полтора-два раза большую производительность на тех же задачах, что и Itanium. Он является, фактически, первым представителем архитектуры IA-64.

Дальше планируется еще большее распараллеливание максимально модным на сегодняшний день путем: процессор должен будет перейти на два физических ядра, что позволит практически удвоить производительность по достаточно приемлемой цене - по крайней мере, результат получится куда более дешевым, чем если бы того же количества исполнительных модулей, регистров, и т.д., пытались достичь на едином кристалле.

6.2. Athlon 64 AMD

В первую очередь заметим, что процессор Athlon 64 – это именно тот 64-битный процессор для настольных систем, который изначально планировала выпустить AMD. Впоследствии, в свете выхода скоростных процессоров Pentium 4, появления в них 800-мегагерцовой шины и технологии Hyper-Threading, AMD в срочном порядке решила нацелить на рынок настольных систем и однопроцессорный Opteron, дав ему имя Athlon 64 FX. Однако Athlon 64 FX в силу своего серверного происхождения оказался дорогим и малораспространенным. По настоящему же продвинуть архитектуру AMD64 для массового использования должен именно Athlon 64.

Ниже представлена таблица 3 со спецификациями 64 - разрядных МП Athlon 64 3200+, Athlon 65 FX-51 и Athlon XP 3200+:

Таблица 3

Характеристика	Athlon 64 FX-51	Athlon 64 3200+	Athlon XP 3200+
Частота	2.2 ГГц	2.0 ГГц	2.2 ГГц
Технология производства	0.13 мкм	0.13 мкм	0.13 мкм
Число транзисторов	105.9 млн.	105.9 млн.	54.3 млн.
Площадь ядра	193 кв.мм	193 кв.мм	101 кв.мм

Номинальное напряжение	1.5В	1.5В	1.65В
Встроенный контроллер памяти	Двуканальный, 128-битный	Одноканальный, 64-битный.	Нет
Поддержка ECC	+	+	-
L1	128 Кбайт (по 64 Кбайта на код и данные)	128 Кбайт (по 64 Кбайта на код и данные)	128 Кбайт (по 64 Кбайта на код и данные)
L2	1024 Кбайт (экс-клюдивный)	1024 Кбайт (экс-клюдивный)	512 Кбайт (экс-клюдивный)

* Заметим, что память в Athlon 64 и Athlon 64 FX тактуется относительно частоты ядра, поэтому реальные частоты для памяти в этом случае составляют 129.4, 157.1 и 200 МГц.

Фактически, Athlon 64 отличается от своего старшего собрата Athlon 64 FX, помимо формы и размеров корпуса, только лишь контроллером памяти. Хотя, при этом, и тот и другой процессоры производятся из одних и тех же кристаллов. Контроллер памяти в Athlon 64 одноканальный и в этом заключается как его слабость, так и преимущество по сравнению с Athlon 64 FX. Недостаток одноканального контроллера памяти в Athlon 64 очевиден: это более низкая теоретическая пропускная способность.

Учитывая, что Athlon 64 способен работать с DDR400 памятью, максимальная пропускная способность встроенного в CPU контроллера памяти составляет 3.2 Гбайт в секунду. Это в два раза меньше, чем аналогичная характеристика Athlon 64 FX. Преимущество же контроллера памяти Athlon 64 заключается в том, что он, в отличие от контроллера Athlon 64 FX, поддерживает обычные нерегистровые модули памяти. Такие модули по сравнению с регистровыми более дешевы, имеет более агрессивные тайминги и работают быстрее, даже при одинаковых с регистровыми модулями настройках. То есть при более низкой пропускной способности, обеспечиваемой контроллером памяти Athlon 64, подсистема па-

мента, его использующая, имеет более низкую латентность, что мы и покажем ниже.

AMD Athlon 64 по внешнему виду похож на Opteron и Athlon 64 FX.



Отличия обнаруживаются только лишь в маркировке и в меньшем числе ножек на обратной стороне, поскольку процессоры Athlon 64 устанавливаются в материнские платы с Socket 754 и не совместимы с Socket 940 платами, предназначенными для CPU семейств Athlon 64 FX и Opteron.

Помимо перечисленных выше особенностей, есть в новых процессорах Athlon 64 и еще одна. Эти процессоры обладают поддержкой технологии Cool'n'Quiet, фактически пришедшей в них из мобильных вариантов МП. По сути, Cool'n'Quiet представляет собой некое подобие технологии энергосбережения PowerNow!, уже давно используемой в мобильных МП от AMD. Но теперь эта технология, наконец, пришла и в настольные процессоры компании. Поддержка Cool'n'Quiet – еще одно преимущество Athlon 64 над Athlon 64 FX/Opteron, не имеющих пока никаких подобных технологий. Компания AMD достаточно давно уделяет пристальное внимание понижению уровня тепловыделения своих настольных процессоров.

Надо сказать, что в этом компания уже давно превосходит Intel: старшие модели процессоров AMD при максимальной нагрузке выделяют значительно меньше тепла, чем старшие модели Pentium 4. Также, в процессорах применяются технологии, понижающие тепловыделение и при низкой нагрузке. Еще МП семей-

ства Athlon XP имели возможность перехода в «ждущий режим» (Halt/Stop Grant) при выполнении команды HALT, что выливалось в понижение температуры процессора при его загрузке ниже 100%. Однако теперь AMD пошла еще дальше. В новых процессорах Athlon 64 реализована еще более интеллектуальная схема понижения тепловыделения.

В дополнение к состояниям Halt/Stop Grant, Athlon 64 умеет сбрасывать свою тактовую частоту и напряжение питания для еще более сильного снижения тепловыделения. В работе с использованием этой технологии тактовой частотой МП управляет драйвер процессора, который сбрасывает или повышает ее, основываясь на данных о его загрузке. Действительно, если процессор полностью справляется с возлагаемой на него работой и его загрузка сильно меньше 100%, то можно без ущерба для функционирования системы в целом снизить его тактовую частоту: на работе системы это никак не скажется. Например, при простоях, работе в офисных приложениях, просмотре видео, дефрагментации дисков и в подобных задачах мощности процессора в полной мере не используются. Именно в таких случаях процессорный драйвер переводит Athlon 64 на меньшую тактовую частоту. Когда же от процессора требуется полная отдача, например, в играх, при решении вычислительных задач, в задачах кодирования данных и т.п., частота процессора поднимается до номинала. Именно таким образом и работает технология Cool'n'Quiet.

На практике это выглядит следующим образом. В обычных условиях, при минимальной загрузке МП процессорный драйвер сбрасывает частоту Athlon 64 3200+ со штатных 2 ГГц до 800 МГц. Напряжение питания процессора при этом понижается до 1.3В. Как видим, снижение тактовой частоты обеспечивается за счет уменьшения множителя процессора до 4х. Это, кстати, обуславливает и тот факт, что процессоры Athlon 64 3200+ поставляются с незафиксированным коэффициентом умножения. В таком режиме процессор продолжает работать до тех пор, пока его загрузка не превысит 70-80%. В частности, мы смогли запустить одновременно дефрагментацию диска, проигрывание файлов с расширением mp3 (аудиофайлов) и просмотр MPEG-4 (видеофайлов)

ролика, в то время как процессор продолжал работать на частоте 800 МГц.

Когда же загрузка процессора Athlon 64 при частоте 800 МГц превышает допустимый предел, МП переводится драйвером в следующее состояние, при котором частота Athlon 64 3200+ составляет 1.8 ГГц, а напряжение питания 1.4В. Достигается это вновь за счет уменьшения множителя, на этот раз до 9х. И только если в данном случае нагрузка процессора вновь оказывается чрезмерно высокой, драйвер переводит МП в штатный режим: частота 2 ГГц, напряжение питания – 1.5В.

Отметим, что в режимах с пониженным питанием и частотой тепловыделение процессора Athlon 64 3200+ резко падает. Для сравнения приведем таблицу 4 с тепловыделением этого процессора в основных режимах.

Таблица 4

Частота	2000 МГц	1800 МГц	800 МГц
Напряжение	1.5 В	1.4 В	1.3 В
Типичное тепловыделение	89 Вт	66 Вт	35 Вт
Типичное тепловыделение в состоянии Halt/Stop Grant	2.2 Вт	2.2 Вт	2.2 Вт

Таким образом, использование технологии Cool'n'Quiet позволяет значительно снизить температуру процессора не только в моменты простоя, но и во время выполнения ряда задач, не требующих от МП максимальной производительности. Что немало важно, быстродействие МП в задачах, требовательных к процессорным ресурсам, при этом совершенно не снижается. В итоге, при применении систем охлаждения с вентиляторами с переменной скоростью, использование технологии Cool'n'Quiet может позволить значительно снизить уровень шума.

6.3. UltraSPARC III Sun

Процессор UltraSPARC III создавался для того, чтобы заменить микропроцессоры второго поколения семейства UltraSPARC. Он имеет производительность, в 2-3 раза превышающую показатели 300-мегагерцовых процессоров UltraSPARC II. Помимо этого, архитектура процессора разрабатывалась с учетом использования его в масштабируемых многопроцессорных системах.

UltraSPARC III представляет собой высокопроизводительный суперскалярный микропроцессор, предназначенный для использования в мощных рабочих станциях и серверах. Он ориентирован на работу в масштабируемых микропроцессорных комплексах.

На рис.18 приведена структурная схема процессора. В его состав входят следующие основные функциональные блоки:

- шесть исполнительных устройств (4 целочисленных и 2 плавающих);
- кэш-памяти команд и данных объемом 32 КБ и 64 КБ соответственно;
- кэш предвыборки и записи объемом по 2 КБ каждый;
- системный интерфейс;
- контроллер основной памяти;
- контроллер L2;
- таблица тэгов L2.

Функции исполнительных устройств распределяются следующим образом. Целочисленные: 2 АЛУ (Integer ALU, ALU/Load), специализированное исполнительное устройство (Load/Store) и устройство обработки ветвлений (Branch Unit). Плавающие: устройства плавающей арифметики/графики (FP adder, FP mul/div/sqrt). Integer ALU, ALU/Load отвечают за выполнение целочисленных операций. Блок Integer ALU содержит фактически два устройства, каждое из которых может выполнять все однотактные операции. Блок ALU/Load выполняет многотактные команды (умножение, деление) и рассчитывает адреса в командах предварительной выборки. В этот блок команды поступают на выполнение по одной, в то время как всего микропроцессор может

выдавать на выполнение до четырех целочисленных команд за такт, одна из которых должна быть командой загрузки или условной пересылки. Функциональных устройств для операций с плавающей запятой тоже два; одна операция сложения и одна операция умножения могут выдаваться на выполнение в каждом такте. В этих устройствах выполняются также и "графические" команды UltraSPARC III.

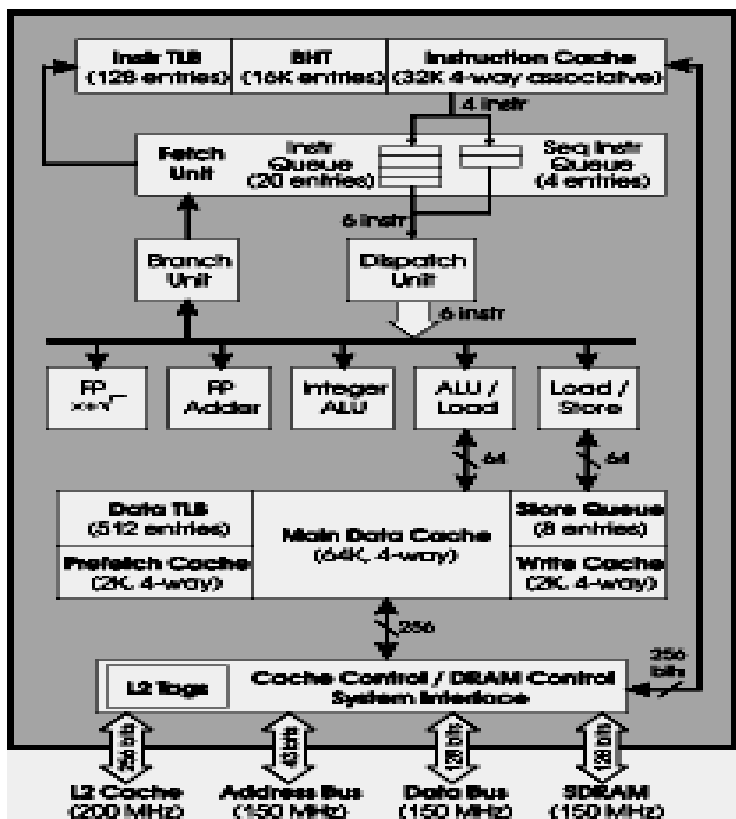


Рис. 18

В UltraSPARC III предусмотрены четырехканальный частично-ассоциативный D-кэш (кэш данных) первого уровня емкостью 64 Кбайт и четырехканальный частично-ассоциативный I-

кэш (кэш инструкций) первого уровня емкостью 32 Кбайт; оба кэш неблокирующиеся.

Следует обратить внимание на большую емкость D-кэш. Кэш данных первого уровня имеет двухтактный доступ, что обусловлено, по всей видимости, высокой частотой процессора (600 МГц).

Кроме кэш-памяти первого уровня, в UltraSPARC III имеется также кэш предварительной выборки (prefetch) емкостью 2 Кбайт, также являющийся четырехканальным частично-ассоциативным. В нем может поместиться до восьми результатов запросов на предварительную выборку. Использование механизма предварительной выборки позволяет избежать блокировки кэш первого уровня, которая могла бы возникнуть из-за превышения порога для числа непопаданий в кэш. Кэш предварительной выборки поддерживает одновременное выполнение двух команд загрузки регистров с общей задержкой всего три такта; за 14 тактов процессора из кэш второго уровня в кэш предварительной выборки можно записать 64 байт данных.

В обоих кэш первого уровня содержатся также буферы быстрой переадресации (I-TLB и D-TLB соответственно), ускоряющие преобразование 64-разрядных виртуальных адресов в 43-разрядные физические. Емкость I-TLB составляет 128 строк, а D-TLB - 512 строк. Кроме 128-строчного буфера, I-TLB включает еще 16-строчный полностью ассоциативный буфер. Надо сказать, что в последнее время разработчики RISC-процессоров уделяют больше внимания буферам TLB; увеличение их емкости предусмотрено архитектурами HP PA-8x00, SGI/MIPS R1x000 и Alpha. В процессоре Alpha 21264 буферы I-TLB и D-TLB имеют емкость по 128 строк.

Еще один тип кэш-памяти UltraSPARC III - это четырехканальный частично-ассоциативный кэш записи емкостью 2 Кбайт. В рассматриваемом микропроцессоре команды записи в память перед выполнением помещаются в очередь записи длиной восемь строк. Запись может производиться либо в D-кэш первого уровня, либо в память более низкого уровня. В последнем случае данные как раз и помещаются сперва в кэш записи. В отличие от D-кэш,

использующего алгоритм сквозной записи, здесь применяется алгоритм обратной записи.

Применение очереди записи и соответствующего кэш способствует объединению нескольких запросов на запись в более крупные блоки. Кроме того, если данные, которые еще находятся в очереди записи, необходимо загрузить в регистр, они загружаются прямо из очереди. В результате, по оценкам Sun, график записи в кэш второго уровня удастся сократить на порядок.

Теги кэш второго уровня в UltraSPARC III расположены на основной микросхеме процессора. Как считают разработчики Sun Microsystems, если бы память тегов размещалась непосредственно на микросхеме кэш-памяти, то задержка при непопадании в кэш второго уровня возросла бы на 10 тактов, а пропускная способность уменьшилась втрое.

Память тегов кэш второго уровня в UltraSPARC III достаточна для поддержки кэш емкостью до 8 Мбайт. Для современных процессоров это максимальная величина. Однако у интеграции в микропроцессор памяти тегов есть и обратная сторона: память тегов занимает на кристалле почти такую же площадь, как I-кэш первого уровня.

Основу архитектуры любого процессора составляет конвейер. В UltraSPARC III увеличено число стадий конвейера до 14 (рис.19).

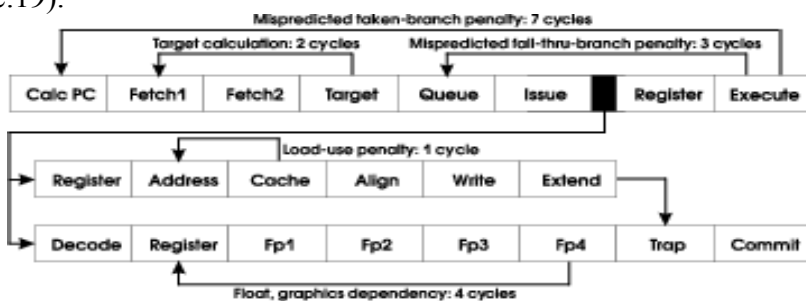


Рис. 19

Это один из самых длинных конвейеров среди всех современных универсальных микропроцессоров архитектуры RISC. Общеизвестное слабое место подобного подхода - проблема заполнения длинных конвейеров. Так, связанная с неправильным

предсказанием перехода задержка в UltraSPARC III составляет 7 тактов. Почти половина длины конвейера (6 ступеней) отводится на подготовку к выполнению команд, столько же - на исполнение команд; две последние ступени - завершающие.

Исполнительная часть конвейера состоит из двух частей: целочисленной и плавающей. Обе части имеют одинаковую длину, что упрощает согласование их работы (позволяет выдавать результаты вычислений в порядке их запуска на исполнение). Аналогичное решение (выровненные конвейеры целочисленной и плавающей арифметики) успешно использовалось в предшествующих поколениях микропроцессоров UltraSPARC.

Большое количество ступеней конвейера можно объяснить существенным повышением тактовой частоты процессора. Более короткие фазы выполнения позволяют избежать длинных связей на кристалле, которые при столь малых технологических нормах начинают вносить заметную дополнительную задержку. Те операции, которые не вписываются в один такт, как, например, выборка команд из кэш-памяти, разбиваются на более мелкие процедуры и выполняются за 2 такта.

Следует отметить еще одну особенность, связанную с исполнительной частью целочисленного конвейера. Для выполнения целочисленных команд отводится 4 такта, реально на это уходит меньше времени (команды АЛУ выполняются за 1 такт). Однако, выровненный конвейер позволяет получить на выходе результаты в том порядке, в котором они поступили на исполнение.

В отличие от многих других современных процессоров, использующих механизм произвольного запуска команд на исполнение, то есть не в порядке их расположения в программе, UltraSPARC III действует строго по порядку. По утверждениям разработчиков, это позволяет сократить объем логики управления в ядре процессора, которая при разработке ее топологии требует большого объема ручной работы, что благоприятно сказывается на быстродействии.

Таким образом, в новом процессоре, как и в его предшественниках, запуск команд на исполнение происходит в порядке их расположения в программе. Процессор позволяет одновременно запустить на исполнение до шести команд за такт (по числу ис-

полнительных устройств); правда, средневзвешенное число одновременно запускаемых команд равно четырем.

Команды выбираются из I-кэш первого уровня и помещаются в очередь команд (буфер) длиной 20 строк (Instruction Queue), откуда группами направляются в соответствующие исполнительные устройства. Максимальное число команд в группе - 6. Все команды в группе получают идентификационный код, в соответствии с которым на выходе из конвейера будут сложены их результаты.

Выше уже упоминалось, что время выполнения большинства целочисленных команд существенно меньше длины исполнительной части целочисленного конвейера, и что фиксированная длина конвейера обеспечивает сохранение порядка поступления команд. Однако, это не означает, что результаты становятся доступны только по достижению выхода конвейера. На самом деле полученные данные могут быть использованы другими командами, находящимися на стадии выполнения, уже на следующем такте после получения результата. Это возможно благодаря наличию рабочего регистрового файла, в котором хранятся все промежуточные результаты вычислений и из которого, по завершению исполнения группы команд, эти результаты переписываются в архитектурный регистровый файл. Таким образом, реальная эффективность конвейера заметно повышается.

Устройство предсказания ветвлений является непременным атрибутом любого современного высокопроизводительного микропроцессора, однако, в каждом конкретном случае оно реализовано по-своему. Основная задача данного устройства состоит в сокращении накладных расходов из-за нарушений в работе конвейера при ветвлении программ. Если взглянуть на конвейер UltraSPARC III, то хорошо видно, сколько придется заплатить за неправильно предсказанный переход. Штрафные санкции составят 7 дополнительных тактов (кстати, у Alpha 21264 — столько же при меньшей длине конвейера).

При решении данной задачи разработчики процессора решили использовать достаточно простой одноуровневый механизм (в отличие от двухуровневого адаптивного механизма в Alpha 21264). Он представляет собой таблицу на 16 К значений (ее раз-

мер увеличен), содержащую информацию об уже происшедших ветвлениях и обеспечивает точность предсказаний на уровне 95% на тестах SPEC95, что близко к аналогичным показателям для Alpha 21264 и AMD K6. Надо отметить, что в UltraSPARC III буфер адресов перехода отсутствует, поэтому при выполнении правильно предсказанного перехода возникает дополнительная задержка.

Помимо механизма предсказания ветвлений в процессоре используется еще и стек адресов возврата на 8 значений, запоминающий адреса возврата при вызове подпрограмм, а также очередь последовательных команд (Sequential Instruction Queue), которая хранит до четырех команд, следующих за командой ветвления, но соответствующих альтернативному пути. В случае, когда предсказанное ветвление окажется неверным, команды из этой очереди сразу направляются на исполнение, экономя несколько тактов.

Производительность вычислительной системы зависит от многих факторов, среди которых быстродействие процессора - не самый главный. Очень многое определяет то, как он взаимодействует с другими компонентами системы.

Первое, что сразу обращает на себя внимание, это большое количество внешних интерфейсов у процессора. Их три: 128-разрядный канал обмена с основной памятью (Main Memory), 256-разрядный канал обмена с L2 и 128-разрядный системный интерфейс.

Достаточно очевидно, что такая многошинная архитектура способствует более эффективной работе многопроцессорной системы в целом. Правда, это новое качество недешево обходится (это и дополнительные расходы на разработку нового корпуса с рекордным числом выводов — 1200, и проблемы повышенного энергопотребления как самого процессора, так и микросхем чипсета и т.п.). Рассмотрим подробнее каждый из перечисленных интерфейсов.

Поскольку архитектура процессора подразумевает единоличное владение памятью, то есть отсутствие непосредственного доступа к ней со стороны каких-либо других устройств, появляется возможность достижения максимальной пропускной способности данного канала, для чего предусмотрен накристалльный кон-

тронлер памяти. Такое решение имеет следующие положительные стороны.

Во-первых, отпадает необходимость в дополнительных внешних компонентах, то есть упрощается сопряжение процессора и памяти. Это к тому же приводит еще и к снижению стоимости.

Во-вторых, возрастает пропускная способность, поскольку производительность канала зависит только от параметров памяти. Так, при использовании SDRAM с тактовой частотой 150 МГц пропускная способность составит 2.4 ГБ/с.

По своей реализации данный канал напоминает описанный выше. Однако, здесь есть ряд принципиальных особенностей.

Во-первых, канал имеет более высокую разрядность — 256 бит. На сегодняшний день UltraSPARC III, пожалуй, единственный микропроцессор, имеющий такую широкую шину данных кэш второго уровня. При использовании микросхем статической памяти (SRAM), работающих на частоте 200 МГц, пропускная способность данного канала способна достичь 6.4 ГБ/с. И это не предел.

Во-вторых, по аналогии с накристалльным SDRAM-контроллером канала основной памяти, данный интерфейс имеет накристалльную таблицу тэгов вторичного кэш. Размер таблицы составляет 90 КБ, и этого достаточно для поддержания кэш-памяти объемом до 8 МБ.

Основным достоинством такого решения является то, что работа с таблицей осуществляется на частоте процессора, то есть результат обращения к кэш становится известен гораздо раньше, чем в случае внекристалльного расположения таблицы тэгов. Соответственно, при непопадании в кэш процедура инициализации обращения к основной памяти начинается на несколько тактов раньше. Аналогично обстоит дело и с поддержкой когерентности кэш в многопроцессорных системах.

Канал записи состоит из трех основных частей: очереди на 8 слов (Store Queue), кэш-памяти данных первого уровня (L1 Data Cache) и кэш-памяти записи (Write Cache). Сразу же отметим, что кэш имеют различные механизмы обновления: L1 кэш данных - сквозной записи, а кэш записи - отложенный. Далее будет понятно, зачем это нужно.

Сначала сохраняемая информация записывается в очередь. Это происходит во время выполнения команды сохранения. Затем, после завершения команды, данные записываются в L1 и, одновременно, в кэш записи. При этом, если происходит непопадание в L1, то его содержимое не обновляется. В противном случае из-за сквозного режима обновления данной кэш-памяти происходило бы постоянное обращение ко вторичному кэш. Таким образом, кэш-память записи как бы дополняет и дублирует L1, но только в процессе записи.

По утверждениям разработчиков, использование такой организации канала записи позволяет сократить трафик на шине вторичной кэш-памяти на 90%.

Системный интерфейс по своим характеристикам аналогичен каналу основной памяти. Из специфических механизмов, свойственных только ему, следует отметить поддержку многопроцессорности (до четырех процессоров в конфигурации с общей шиной и более четырех при иерархической структуре шин).

В настоящее время фирмой Sun предлагается множество вариантов UltraSparc, как одного из самого успешных процессоров с RISC-архитектурой. Этот процессор можно найти во встраиваемых системах высших моделей, но его основное назначение – высокопроизводительные рабочие станции и серверы. Архитектура процессоров этого семейства использует регистровые окна, а не стек блоков памяти, чем подобна EPIC-архитектуре фирмы Intel.

Последняя модель семейства – UltraSpark IV – первый представитель процессоров, выполненных на основе концепции "организации многопоточной архитектуры на уровне микросхемы" (Chip Multithreading – CMT) в рамках инициативы "производительные вычисления" (Throughput Computing). Процессор UltraSpark IV совместим на уровне двоичных кодов со Spark-процессорами предыдущих поколений и поддерживает 8 и 9 версии ОС Solaris v.8 и 9.

UltraSpark IV содержит двухпроцессорное ядро на базе конвейерной (глубина конвейера каждого ядра – 14 стадий) архитектуры процессора предыдущего поколения UltraSpark III. 16-входной буфер команд выдает конвейеру на выполнение четыре команды/такт, а каждый суперскалярный процессор ядра выполня-

ет четыре команды/такт. Шести исполнительным устройствам параллельного действия (двум целочисленным устройствам, одному выполнения условного перехода, одному устройству загрузки регистров/записи в память и двум функциональным устройствам с плавающей точкой) выдаются на выполнение шесть команд за такт.

L1 кэш каждого ядра содержит 64-Кбайт памяти данных, 32-Кбайт – команд, 32 Кбайт – записи в память и 2 Кбайт – предварительной выборки. Как и все современные 64-бит микропроцессоры, UltraSpark IV имеет внутрисхемный контроллер внешней оперативной памяти (статического ДОЗУ). Нити (процессы), выполняемые UltraSpark IV, совместно используют адресную шину и шину данных для получения доступа к ячейкам L2 кэш (типа СОЗУ) объемом 8 Мбайт, к контроллеру оперативной памяти и соединительной шине типа Sun Fireplane с максимальной пропускной способностью 2,4 Гбайт/с. Максимальная мощность, потребляемая схемой, составляет 108 Вт при напряжении питания 1,35 В и частоте 1,2 ГГц.

Выполнен микропроцессор по 0,13-мкм КМОП-технологии с семислойной медной металлизацией фирмы Texas Instruments (с которой Sun решает проблемы производства разрабатываемых изделий). Он содержит 66 млн. транзисторов. Монтируется в 1368-выводной типа LGA.

Ключ к успеху процессоров семейства UltraSpark – операционная система Solaris фирмы Sun Microsystems. Чтобы в дальнейшем реализовывать все возможности этой ОС на более высоком уровне, необходимо совершенствовать поддерживающее ее "железо". И фирма не отказывается от этой задачи, ставя целью создание процессоров с более радикальной многопоточной архитектурой, более чем в 30 раз превосходящих по быстродействию современные процессоры для корпоративных систем.

Фирмой планируется создание процессора (кодовое название Niagara) с восемью процессорными ядрами на кристалле, одновременно обрабатываемыми в целом до 32 нитей. Можно с уверенностью сказать, что компания Sun Microsystems хорошо понимает, что понадобится в ближайшие годы ей и ее заказчикам.

6.4. Alpha 21264 DEC

Развитие МП Alpha сначала шло в направлении первоочередного роста тактовой частоты при относительно простой микроархитектуре МП. Наиболее ярко это выразилось в Alpha 21064. Микроархитектура 21164 сильно усложнилась, а сам МП далеко обогнал всех оппонентов по тактовой частоте и пиковой производительности при работе с вещественной арифметикой. При этом этот процессор имеет существенно более простое внутреннее строение, чем суперскалярные микропроцессоры с внеочередным выполнением команд - HP PA-8x00 или SGI/MIPS R10000. Однако, в Alpha 21264 разработчики также обратилась к внеочередному суперскалярному выполнению команд.

Это подтверждает, что наиболее перспективна сложная микроархитектура МП, а не простой суперконвейерный подход, при котором проще поднимать тактовую частоту. Из современных процессоров RISC, если не считать IBM P2SC с сильно отличным от других строением, только UltraSPARC и Alpha 21164 не поддерживают внеочередное выполнение команд. С другой стороны, нельзя сказать, что компания DEC шла по неверному пути, когда стремилась в первую очередь увеличивать тактовую частоту, сохраняя относительно более простую архитектуру МП Alpha. Доказательством этого служит тот факт, что Alpha все это время была лидером производительности.

Микроархитектура Alpha 21264 представлена на рис.20.

Собственно, ключевые особенности, с которыми "играют" разработчики современных высокопроизводительных RISC-процессоров, относятся к трем основным областям: кэш-память; внеочередное выполнение в ФИУ; алгоритмы предсказания переходов.

Очевидно, что при высоких тактовых частотах возникают проблемы как для внутренней, так и для внешней кэш-памяти. Что касается внутреннего кэш, то в Alpha 21164 это привело к созданию уникального интегрированного на чипе двухуровневой кэш, включающей прямоадресуемые I- и D-кэш L1 емкостью 8 Кбайт каждый, плюс 3-канальный частично-ассоциативный кэш L2 емкостью 96 Кбайт.

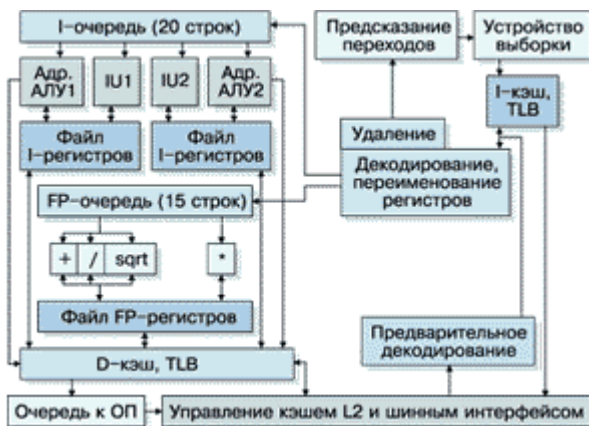


Рис. 20

Подобную конструкцию не повторили ни в одном из других известных микропроцессоров. Размер L1 ограничен требованиями доступа к нему за 2 нс (для 21164/500 МГц). Процент непопаданий в такой кэш относительно велик (следует учесть, что он прямоадресуемый, что несколько увеличивает вероятность непопадания по сравнению с многоканальным частично-ассоциативным кэш), а для обращения в L2 требуется уже 6 тактов. Кроме L1 и L2, для достижения высокой производительности применяется внешний кэш L3 большой емкости.

В Alpha 21264 применены 2-канальные частично-ассоциативные I- и D-кэш емкостью 64 Кбайт каждый. Очевидно, что вероятность попадания в такой кэш гораздо выше. Элиминируется и обмен данными между кэш внутри чипа 21164. Емкость этого кэш в 2 раза больше, чем в SGI/MIPS R10000. Наибольшую емкость кэш на чипе сегодня имеет IBM P2SC - 128 Кбайт данных плюс 32 Кбайт для команд. Однако этот микропроцессор вообще не имеет вторичного кэш.

Микропроцессоры DEC имеют непосредственно взаимодействующие блоки микропроцессора, располагающиеся рядом. D-кэш должен работать как с целочисленными ФИУ и ФИУ с плавающей запятой, так и с системной шиной. За увеличение емкости первичного кэш в 21264 пришлось заплатить увеличением време-

ни доступа до 2 тактов. Двухтактный кэш L1 используется и в HP PA-8000, но в качестве внешнего.

Очевидно, что увеличение в Alpha 21264 по сравнению с Alpha 21164 на 1 такт времени доступа в кэш L1 полностью компенсируется его большей емкостью и одноуровневой организацией. DEC оценивает уменьшение общей пиковой производительности из-за этого дополнительного такта в 4%. Альтернативные подходы привели бы к большему ее снижению.

D-кэш L1 в Alpha 21264 является двухпортовым и позволяет получать 2 независимых 64-разрядных результата за такт. Это достигается путем запуска нового доступа каждые полтакта - D-кэш работает на частоте 1 ГГц (в предположении 500 МГц частоты самого 21264). До 8 непопаданий в кэш L1 чипа 21264 не блокируют его работу.

В отличие от Alpha 21164, в Alpha 21264 внешний кэш L2 использует выделенную (собственную) шину, подобно MIPS R10000, HP PA-8000/8200 или Intel Pentium Pro/Pentium II. Это очень важное усовершенствование. В Alpha 21164 внешний кэш применяет для передачи данных 128-разрядную системную шину, что повышает нагрузку на шину в многопроцессорных системах и может отрицательно сказываться на масштабируемости. Кэш L2 в 21264 может работать на частотах 2/3, 1/2, 1/3 и 1/4 от тактовой частоты процессора, но не выше 333 МГц.

Общей проблемой высокочастотных микропроцессоров на сегодня является отсутствие микросхем SRAM для кэш с большой тактовой частотой. Дешевые системы на базе 21264 будут использовать 133 МГц-кэш SRAM, подобный применяемому в Intel Pentium II/266. Однако пропускная способность L2 в Alpha 21264 в 2 раза выше (до 2.1 Гбайт/с), чем у Pentium II - разрядность шины равна 128 против 64 в Pentium II. К моменту начала поставок Alpha 21264 были выпущены схемы SRAM с частотой 200-250 МГц, чему отвечает пропускная способность кэша 3.2-4.0 Гбайт/с. Отметим, что SGI/MIPS R10000 работает с 200 МГц внешним кэшем как раз на 128-разрядной выделенной шине.

DEC совместно с фирмами Motorola и Samsung участвовала в разработке нового типа SRAM - "dual data". Они аналогичны 5нс- SRAM, но за 1 такт через выходные шины отсылают данные

сразу в 2 транзакции. Работа на частоте 167 МГц в этом случае эквивалентна 333 МГц-передаче данных. Такой кэш позволяет достигнуть максимальной для 21264 пропускной способности - 5.3 Гбайт/с. Большее значение, 5.8 Гбайт/с, имеет HP PA-8000.

Использование дешевой кэш-памяти с частотой в 1/4 от процессорной понижает производительность при работе с целыми числами на 5%, а с плавающей точкой - не менее чем на 20 %. Задержка при обращении во внешний кэш составляет 12 тактов, а для 133 МГц кэш - 14 тактов. При этом, как и внутренний кэш, кэш L2 является неблокирующимся.

Одновременно с доступом в D-кэш L1 в Alpha 21264 осуществляется преобразование виртуальных адресов в физические с использованием буфера TLB емкостью 128 строк, что в 2 раза больше, чем в SGI/MIPS R10000; HP PA-8000 имеет TLB на 96 строк, PA-8200 - на 120 строк.

В 21264 применено внеочередное выполнение команд (не в соответствии с порядком команд в коде программы) и переименование регистров на лету. Если Alpha 21164 имеет 32 целочисленных регистра и 32 регистра с плавающей запятой, то в 21264 есть 80 целочисленных физических регистров (имеется даже 2 копии этого файла) и 72 физических регистра с плавающей запятой. Внеочередное выполнение команд может иметь место как для очереди целочисленных команд, так и для очереди команд с плавающей запятой.

Чип Alpha 21264 способен выполнять до 6 команд за такт (поддерживаемый уровень - 4 команды за такт). В это число включены также команды загрузки регистров/записи в память. Команды выбираются из I-кэша, куда они попадают частично декодированными. Предварительное декодирование создает 3-разрядное поле, по которому определяется, к какому ФИУ следует направить команду: целочисленному ФИУ общего назначения, (целочисленному) адресному АЛУ или ФИУ с плавающей запятой. Это 3-разрядное поле хранится вместе с командами в I-кэше.

Alpha 21264 имеет 6 ФИУ, из них 4 целочисленных: 2 - общего назначения, и 2 - адресных АЛУ. Последние отвечают за выполнение всех команд загрузки/записи в память и могут также выполнять простые арифметические и логические целочисленные

операции. 2 ФИУ общего назначения производят арифметические и логические операции над целыми числами, операции сдвига и перехода. Одно из них умеет умножать, а другое - выполнять новые команды из мультимедийного расширения набора команд MVI (Motion-Video Instructions).

После окончательного декодирования "целочисленные" команды отправляются в очередь емкостью 20 строк. Простые целочисленные операции, например, сложение, могут выполняться в любых целочисленных ФИУ. С целью упрощения логики уже при постановке в очередь решается, направлять ли команду в целочисленные ФИУ общего назначения или в адресные АЛУ. Каждый такт команды из очереди, операнды которых доступны, арбитражируются на предмет направления в ФИУ. Выборка команд осуществляется не обязательно в соответствии с их расположением в программе.

Одновременно может выполняться 4 операции над целыми числами. Это требует файла целочисленных регистров с 8 портами чтения и 6 портами записи. Вместо этого разработчики по технологическим причинам продублировали файл целочисленных регистров. Каждая копия файла имеет 4 порта чтения и 6 портов записи. Каждому файлу ставится в соответствие свое целочисленное ФИУ общего назначения и АЛУ. Такой комплект оборудования DEC называла "кластером". Файлы регистров остаются синхронизированными, но это требует одного дополнительного такта для записи из АЛУ одного кластера в файл регистров другого кластера. По оценкам DEC, кластеризация понижает Пт в среднем на 1%. Последовательность взаимозависимых команд имеет тенденцию попадать в один и тот же кластер благодаря "сдвигу" доступности операндов на 1 такт.

ФИУ с плавающей запятой в Alpha 21264 всего 2. Их конвейеры имеют длину 4 такта, что больше, чем у HP PA-8000 (3 такта) и MIPS R10000 (2 такта). Представление об основных стадиях конвейеров МП 21264 можно получить из рис. 21. Alpha 21264 может поддерживать одновременное выполнение 4 команд с плавающей запятой за такт в смеси из 50% операций загрузки регистров/записи в память, 25% умножений и 25% сложений. Как и MIPS R10000, Alpha 21264 может выдавать 2 результата с пла-

вающей запятой (FLOP) за такт. Лишь микропроцессоры HP PA-8x00 и IBM Power2/P2SC способны выдавать 4 FLOP за такт. Операции деления и извлечения квадратного корня не конвейеризованы и занимают 16 и 33 такта соответственно.

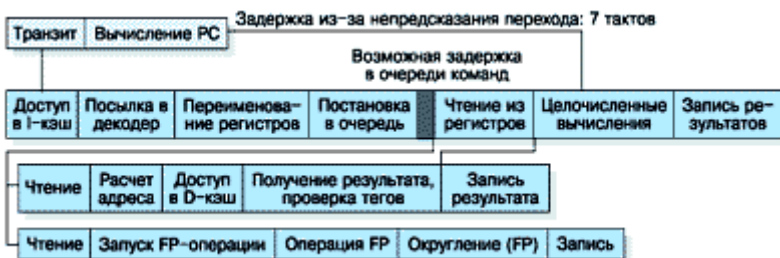


Рис. 21

Важнейшим компонентом современных суперскалярных микропроцессоров является предсказание переходов. Неправильное предсказание заметно ухудшает производительность. Так, средняя задержка при неправильном предсказании в Alpha 21264 составляет свыше 11 тактов. Особенно важным является предсказание переходов для программ, ориентированных на решение коммерческих задач, для которых характерен большой процент команд условного перехода. Алгоритмы предсказания быстро развиваются; одни методы предсказания переходов оказываются точнее других, но разные алгоритмы хорошо работают на разных типах переходов.

Попытка скомбинировать статическое и динамическое предсказания переходов предпринята в HP PA-8500. В Alpha 21264 реализована оригинальная комбинация двух методов динамического предсказания. При этом комбинируются оценки двух таблиц: двухуровневой таблицы "локальных предсказаний" (1024 строки по 10 бит плюс 1024 трехразрядных строки в "сводной" таблице второго уровня), индексом в которой является счетчик команд, и одноуровневой таблицы глобальной истории предсказаний с 2-разрядными полями истории. В третьей таблице ведется "история" обоих предсказаний, на основании которой для каждого

перехода динамически выбирается лучший алгоритм. Эта таблица имеет емкость 4096 двухразрядных полей.

Оценка точности предсказания переходов для тестов SPECint95 составляет порядка 95%. Даже в случае правильно предсказанного перехода в Alpha 21264 из-за двухтактного доступа к I-кэшу приходится предпринимать особые меры по уменьшению задержек, связанных с переходом. Каждая строка в I-кэше содержит 4 команды вместе с дополнительными полями, относящимися к предсказанию, какая строка будет выбираться следующей. Преимуществом 21264 является то, что выборка команд происходит относительно автономно (от работы остальных блоков МП), при этом в декодер направляется по 4 команды за такт.

Подобно многим другим МП, Alpha 21264 имеет стек адресов возврата, применяемый для организации эффективного выхода из вызванных подпрограмм. Этот стек больше, чем в других МП, и позволяет аккуратно отслеживать до 32 уровней вызова подпрограмм.

Минусом микроархитектуры Alpha 21264 является группировка команд в "четверки" и необходимость их соответствующего выравнивания для эффективной обработки переходов. Так, желательно, чтобы адрес перехода указывал на первую команду четверки.

64-разрядная системная шина Alpha 21264 может работать на частотах до 333 МГц, чему соответствует пиковая пропускная способность 2.7 Гбайт/с (поддерживаемое значение - 2.0 Гбайт/с). Это выше, чем в славящихся своей высокой пропускной способностью чипах IBM Power2/P2SC (максимальная пиковая величина - для 77 МГц Power2 - равна 2.5 Гбайт/с). Как и для внешнего кэша, для системной шины могут применяться делители тактовой частоты процессора.

Фактически эта шина осуществляет соединение "точка-точка", что говорит об ориентации на применение коммутаторов вместо традиционной системной шины. Коммутаторы используются в известных многопроцессорных системах: NUMA-системах SGI Origin 2000, серверах серии SPP от Convex/HP и др. Можно предположить, что подобное строение будет применяться и в бу-

дущих мультипроцессорных системах DEC/Compaq с кодовым названием Wildfire, которые должны базироваться на Alpha 21264.

В многопроцессорной системе каждый процессор 21264 имеет свой канал в оперативную память, хотя пропускная способность самой системы памяти разделяется процессорами. Системная шина использует протокол с расщеплением транзакций; до 16 ссылок к памяти могут обрабатываться одновременно. Каждый процессор должен иметь собственное соединение с реализующим интерфейс системной шины набором микросхем "Цунами". Нельзя просто добавить на шину второй процессор - это нарушает связь "точка-точка". Цунами поддерживает соединение с двумя процессорами; возможно создание "4-процессорной" версии.

Набор микросхем Цунами позволяет демультиплексировать 64-разрядные данные шины в 128-, 256- и даже 512-разрядный интерфейс оперативной памяти на SDRAM, а также подсоединять до 2 64-разрядных шин PCI. Ширина тракта к оперативной памяти и число шин PCI определяется количеством используемых микросхем разного типа из набора Цунами.

Alpha 21264 создается на базе 0.35 мкм-технологии и имеет 15.2 млн. Площадь МП - около 302 мм²; рассеяние тепла около 60 ватт при тактовой частоте 500 МГц (около 72 ватт при тактовой частоте 600 МГц)

6.5. PA 7100 Hewlett-Packard

Особенностью архитектуры PA 7100 является внекристальная реализация кэш, что позволяет реализовать различные объемы кэш-памяти и оптимизировать конструкцию в зависимости от условий применения (рис. 22). Хранение команд и данных осуществляется в отдельных кэшах, причем процессор соединяется с ними с помощью высокоскоростных 64-битовых шин.

Кэш-память реализуется на высокоскоростных кристаллах статической памяти (SRAM), синхронизация которых осуществляется непосредственно на тактовой частоте процессора. При тактовой частоте 100 МГц каждый кэш имеет полосу пропускания 800 Мбайт/с при выполнении операций считывания и 400 Мбайт/с при выполнении операций записи.

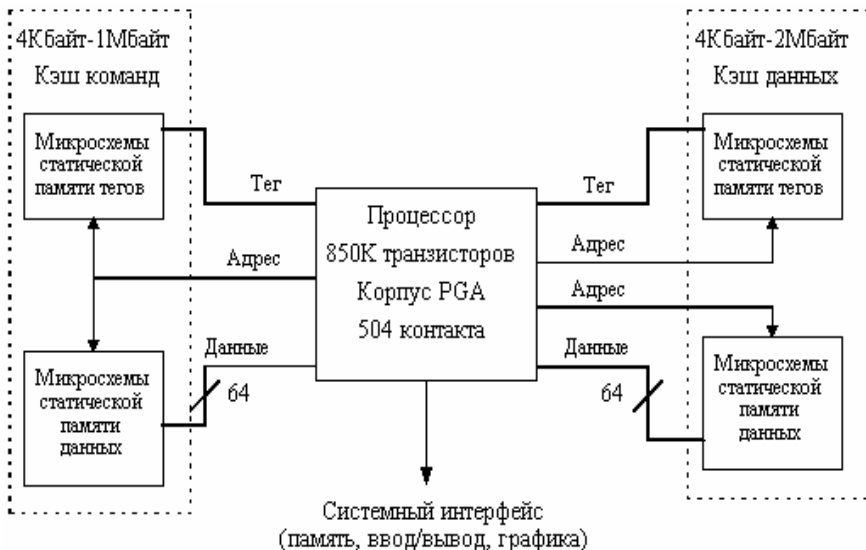


Рис. 22

Микропроцессор аппаратно поддерживает различный объем кэш-памяти: кэш команд может иметь объем от 4 Кбайт до 1 Мбайт, кэш данных - от 4 Кбайт до 2 Мбайт. Чтобы снизить коэффициент промахов применяется механизм хеширования адреса. В обоих кэшах для повышения надежности применяются дополнительные контрольные разряды, причем ошибки кэша команд корректируются аппаратными средствами.

Процессор подсоединяется к памяти и подсистеме ввода/вывода посредством синхронной шины. Процессор может работать с тремя разными отношениями внутренней и внешней тактовой частоты в зависимости от частоты внешней шины: 1:1, 3:2 и 2:1. Это позволяет использовать в системах разные по скорости микросхемы памяти.

Конструктивно на кристалле PA-7100 размещены целочисленный процессор, процессор для обработки чисел с плавающей точкой, устройство управления кэш-памятью, унифицированный буфер TLB, устройство управления, а также ряд интерфейсных схем. Целочисленный процессор включает АЛУ, устройство сдви-

га, сумматор команд перехода, схемы проверки кодов условий, схемы обхода, универсальный регистровый файл, регистры управления и регистры адресного конвейера.

Устройство управления кэш-памятью содержит регистры, обеспечивающие перезагрузку кэш-памяти при возникновении промахов и контроль когерентного состояния памяти. Это устройство содержит также адресные регистры сегментов, буфер преобразования адреса TLB и аппаратуру хеширования, управляющую перезагрузкой TLB.

В состав процессора плавающей точки входят устройство умножения, арифметико-логическое устройство, устройство деления и извлечения квадратного корня, регистровый файл и схемы "закоротки" результата. Интерфейсные устройства включают все необходимые схемы для связи с кэш-памятью команд и данных, а также с шиной данных. Обобщенный буфер TLB содержит 120 строк ассоциативной памяти фиксированного размера и 16 строк переменного размера.

Устройство плавающей точки (рис. 23) реализует арифметику с одинарной и двойной точностью в стандарте IEEE 754.

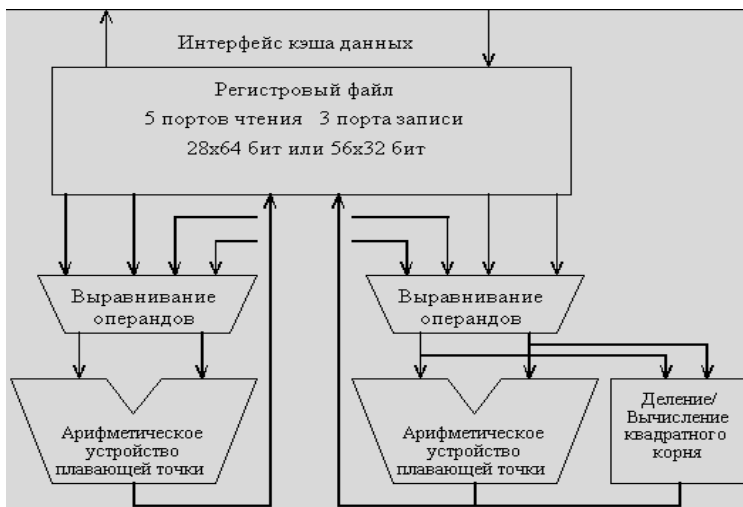


Рис. 23

Его устройство умножения используется также для выполнения операций целочисленного умножения. Устройства деления и вычисления квадратного корня работают с удвоенной частотой процессора. Арифметико-логическое устройство выполняет операции сложения, вычитания и преобразования форматов данных. Регистровый файл состоит из 28 64-битовых регистров, каждый из которых может использоваться как два 32-битовых регистра для выполнения операций с плавающей точкой одинарной точности. Регистровый файл имеет пять портов чтения и три порта записи, которые обеспечивают одновременное выполнение операций умножения, сложения и загрузки/записи.

Большинство улучшений производительности процессора связано с увеличением тактовой частоты до 100 МГц по сравнению с 66 МГц у его предшественника.

Конвейер целочисленного устройства включает шесть ступеней: чтение из кэша команд (IR), чтение операндов (OR), выполнение/чтение из кэша данных (DR), завершение чтения кэша данных (DRC), запись в регистры (RW) и запись в кэш данных (DW). На ступени ID выполняется выборка команд.

Реализация механизма выдачи двух команд требует небольшого буфера предварительной выборки, который обеспечивает предварительную выборку команд за два такта до начала работы ступени IR. Во время выполнения на ступени OR все исполнительные устройства декодируют поля операндов в команде и начинают вычислять результат операции. На ступени DR целочисленное устройство завершает свою работу. Кроме того, кэш-память данных выполняет чтение, но данные не поступают до момента завершения работы ступени DRC.

Результаты операций сложения (ADD) и умножения (MULTIPLY) также становятся достоверными в конце ступени DRC. Запись в универсальные регистры и регистры плавающей точки производится на ступени RW. Запись в кэш данных командами записи (STORE) требует двух тактов. Наиболее раннее двухтактное окно команды STORE возникает на ступенях RW и DW. Однако это окно может сдвигаться, поскольку записи в кэш данных происходят только когда появляется следующая команда записи. Операции деления и вычисления квадратного корня для чи-

сел с плавающей точкой заканчиваются на много тактов позже ступени DW.

Конвейер проектировался с целью максимального увеличения времени, необходимого для выполнения чтения внешних кристаллов SRAM кэш-памяти данных. Это позволяет максимизировать частоту процессора при заданной скорости SRAM. Все команды загрузки (LOAD) выполняются за один такт и требуют только одного такта полосы пропускания кэш-памяти данных. Поскольку кэши команд и данных размещены на разных шинах, в конвейере отсутствуют какие-либо потери, связанные с конфликтами по обращениям в кэш данных и кэш команд.

Процессор может в каждом такте выдавать на выполнение одну целочисленную команду и одну команду плавающей точки. Полоса пропускания кэша команд достаточна для поддержания непрерывной выдачи двух команд в каждом такте. Отсутствуют какие-либо ограничения по выравниванию или порядку следования пары команд, которые выполняются вместе. Кроме того, отсутствуют потери тактов, связанных с переключением с выполнения двух команд на выполнение одной команды. Специальное внимание было уделено тому, чтобы выдача двух команд в одном такте не приводила к ограничению тактовой частоты. Чтобы добиться этого, в кэше команд был реализован специально предназначенный для этого заранее декодируемый бит, чтобы отделить команды целочисленного устройства от команд устройства плавающей точки. Этот бит предварительного декодирования команд минимизирует время, необходимое для правильного разделения команд.

Потери, связанные с зависимостями по данным и управлению, в этом конвейере минимальны. Команды загрузки выполняются за один такт, за исключением случая, когда последующая команда пользуется регистром-приемником команды LOAD. Как правило, компилятор позволяет обойти подобные потери одного такта. Для уменьшения потерь, связанных с командами условного перехода, в процессоре используется алгоритм прогнозирования направления передачи управления. Для оптимизации производительности циклов передачи управления вперед по программе прогнозируются как невыполняемые переходы, а передачи управле-

ния назад по программе - как выполняемые переходы. Правильно спрогнозированные условные переходы выполняются за один такт.

Количество тактов, необходимое для записи слова или двойного слова командой STORE, уменьшено с трех до двух тактов. В более ранних реализациях архитектуры PA-RISC был необходим один дополнительный такт для чтения тега кэш, чтобы гарантировать попадание, а также для того, чтобы объединить старые данные строки кэш-памяти данных с записываемыми данными. PA 7100 использует отдельную шину адресного тега, чтобы совместить по времени чтение тега с записью данных предыдущей команды STORE. Кроме того, наличие отдельных сигналов разрешения записи для каждого слова строки кэш-памяти устраняет необходимость объединения старых данных с новыми, поступающими при выполнении команд записи слова или двойного слова.

Этот алгоритм требует, чтобы запись в микросхемы SRAM происходила только после того, когда будет определено, что данная запись сопровождается попаданием в кэш и не вызывает прерывания. Это требует дополнительной ступени конвейера между чтением тега и записью данных. Такая конвейеризация не приводит к дополнительным потерям тактов, поскольку в процессоре реализованы специальные цепи обхода, позволяющие направить отложенные данные команды записи последующим командам загрузки или командам STORE, записывающим только часть слова. Для данного процессора потери конвейера для команд записи слова или двойного слова сведены к нулю, если непосредственно последующая команда не является командой загрузки или записи. В противном случае потери равны одному такту. Потери на запись части слова могут составлять от нуля до двух тактов.

Моделирование показывает, что подавляющее большинство команд записи в действительности работают с однословным или двухсловным форматом.

Все операции с плавающей точкой, за исключением команд деления и вычисления квадратного корня, полностью конвейеризованы и имеют двухтактную задержку выполнения как в режиме с одинарной, так и с двойной точностью. Процессор может выдавать на выполнение независимые команды с плавающей точкой в

каждом такте при отсутствии каких-либо потерь. Последовательные операции с зависимостями по регистрам приводят к потере одного такта. Команды деления и вычисления квадратного корня выполняются за 8 тактов при одиночной и за 15 тактов при двойной точности. Выполнение команд не останавливается из-за команд деления/вычисления квадратного корня до тех пор, пока не потребуется регистр результата или не будет выдаваться следующая команда деления/вычисления квадратного корня.

Процессор может выполнять параллельно одну целочисленную команду и одну команду с плавающей точкой. При этом "целочисленными командами" считаются и команды загрузки и записи регистров плавающей точки, а "команды плавающей точки" включают команды FMPYADD и FMPYSUB. Эти последние команды объединяют операцию умножения с операциями сложения или вычитания соответственно, которые выполняются параллельно. Пиковая производительность составляет 200 MFLOPS для последовательности команд FMPYADD, в которых смежные команды независимы по регистрам.

Потери для операций плавающей точки, использующих предварительную загрузку операнда командой LOAD, составляют один такт, если команды загрузки и плавающей арифметики являются смежными, и два такта, если они выдаются для выполнения одновременно. Для команды записи, использующей результат операции с плавающей точкой, потери отсутствуют, даже если они выполняются параллельно.

Потери, возникающие при промахах в кэш данных, минимизируются посредством применения четырех разных методов: "попадание при промахе" для команд LOAD и STORE, потоковый режим работы с кэшем данных, специальная кодировка команд записи, позволяющая избежать копирования строки, в которой произошел промах, и семафорные операции в кэш-памяти. Первое свойство позволяет во время обработки промаха в кэш данных выполнять любые типы других команд. Для промахов, возникающих при выполнении команды LOAD, обработка последующих команд может продолжаться до тех пор, пока регистр результата команды LOAD не потребуется в качестве регистра операнда для другой команды.

Компилятор может использовать это свойство для предварительной выборки в кэш необходимых данных задолго до того момента, когда они действительно потребуются. Для промахов, возникающих при выполнении команды STORE, обработка последующих команд загрузки или операций записи в части одного слова продолжается до тех пор, пока не возникает обращений к строке, в которой произошел промах. Компилятор может использовать это свойство для выполнения команд на фоне записи результатов предыдущих вычислений. Во время задержки, связанной с обработкой промаха, другие команды LOAD и STORE, для которых происходит попадание в кэш данных, могут выполняться и другие команды целочисленной арифметики и плавающей точки. В течение всего времени обработки промаха команды STORE, другие команды записи в ту же строку кэш-памяти могут происходить без дополнительных потерь времени. Для каждого слова в строке кэш-памяти процессор имеет специальный индикационный бит, предотвращающий копирование из памяти тех слов строки, которые были записаны командами STORE. Эта возможность применяется к целочисленным и плавающим операциям LOAD и STORE.

Выполнение команд останавливается, когда регистр-приемник команды LOAD, выполняющейся с промахом, требуется в качестве операнда другой команды. Свойство "потоковости" позволяет продолжить выполнение, как только нужное слово или двойное слово возвращается из памяти. Таким образом, выполнение команд может продолжаться как во время задержки, связанной с обработкой промаха, так и во время заполнения соответствующей строки при промахе.

При выполнении блочного копирования данных в ряде случаев компилятор заранее знает, что запись должна осуществляться в полную строку кэш-памяти. Для оптимизации обработки таких ситуаций архитектура определяет специальную кодировку команд записи ("блочное копирование"), которая показывает, что аппаратуре не нужно осуществлять выборку из памяти строки, при обращении к которой может произойти промах кэш-памяти. В этом случае время обращения к кэшу данных складывается из времени, которое требуется для копирования в память старой строки кэш-памяти по тому же адресу в кэш (если он "грязный") и

времени, необходимого для записи нового тега кэш. В процессоре РА 7100 такая возможность реализована как для привилегированных, так и для непривилегированных команд.

Последнее улучшение управления кэш-памятью данных связано с реализацией семафорных операций "загрузки с обнулением" непосредственно в кэш-памяти. Если семафорная операция выполняется в кэш, то потери времени при ее выполнении не превышают потерь обычных операций записи. Это не только сокращает конвейерные потери, но и снижает трафик шины памяти. В архитектуре предусмотрен также другой тип специального кодирования команд, который устраняет требование синхронизации семафорных операций с устройствами ввода/вывода.

Управление кэш-памятью команд позволяет при промахе продолжить выполнение команд сразу же после поступления отсутствующей в кэше команды из памяти. 64-битовая магистраль данных, используемая для заполнения блоков кэша команд, соответствует максимальной полосе пропускания внешней шины памяти 400 Мбайт/с при тактовой частоте 100 МГц.

В процессоре предусмотрен также ряд мер по минимизации потерь, связанных с преобразованиями виртуальных адресов в физические.

Конструкция процессора обеспечивает реализацию двух способов построения многопроцессорных систем. При первом способе каждый процессор подсоединяется к интерфейсному кристаллу, который наблюдает за всеми транзакциями на шине основной памяти. В такой системе все функции по поддержанию когерентного состояния кэш-памяти возложены на интерфейсный кристалл, который посылает процессору соответствующие транзакции. Кэш данных построен на принципах отложенного обратного копирования и для каждого блока кэш-памяти поддерживаются биты состояния "частный" (private), "грязный" (dirty) и "достоверный" (valid), значения которых меняются в соответствии с транзакциями, которые выдает или принимает процессор.

Второй способ организации многопроцессорной системы позволяет объединить два процессора и контроллер памяти и ввода-вывода на одной и той же локальной шине памяти. В такой конфигурации не требуется дополнительных интерфейсных кри-

сталлов и она совместима с существующей системой памяти. Когерентность кэш-памяти обеспечивается наблюдением за локальной шиной памяти. Пересылки строк между кэшами выполняются без участия контроллера памяти и ввода-вывода. Такая конфигурация обеспечивает возможность построения очень дешевых высокопроизводительных многопроцессорных систем.

Процессор поддерживает ряд операций, необходимых для улучшения графической производительности рабочих станций серии 700: блочные пересылки, Z-буферизацию, интерполяцию цветов и команды пересылки данных с плавающей точкой для обмена с пространством ввода/вывода.

Процессор построен на базе технологического процесса КМОП с проектными нормами 0.8 микрон, что обеспечивает тактовую частоту 100 МГц.

6.6. R12000 MIPS

Организация памяти микропроцессора R12000. MIPS R12000 использует два уровня кэш-памяти. Первый уровень наборно ассоциативной кэш-памяти, расположенной на кристалле, имеет 32КБ для данных и 32КБ для команд. Второй уровень - внешняя кэш-память объединена для данных и для команд и может иметь размер от 512КБ до 16МБ.

При разработке процессора R12000 большое внимание было уделено эффективной реализации иерархии памяти. В данном чипе обеспечиваются раннее обнаружение промахов кэш-памяти и параллельная перезагрузка строк с выполнением другой полезной работой. Реализованные на кристалле кэш-памяти поддерживают одновременную выборку команд, выполнение команд загрузки и записи данных в память, а также операций перезагрузки строк кэш-памяти. Заполнение строк кэш-памяти выполняется по принципу "запрошенное слово первым", что позволяет существенно сократить простои процессора из-за ожидания требуемой информации. Все кэш имеют двухканальную множественно ассоциативную организацию с алгоритмом замещения LRU.

Кэш-память данных первого уровня процессора R12000 имеет емкость 32 Кбайт и организована в виде двух одинаковых

банков емкостью по 16 Кбайт, что обеспечивает двукратное расслоение при выполнении обращений к этой кэш-памяти. Каждый банк представляет собой двухканальную множественно-ассоциативную кэш-память с размером строки (блока) в 32 байта. Кэш данных индексируется с помощью виртуального адреса и хранит теги физических адресов памяти. Такой метод индексации позволяет выбрать подмножество кэш-памяти в том же такте, в котором формируется виртуальный адрес. Однако для того, чтобы поддерживать когерентность с кэш-памятью второго уровня, в L1 хранятся теги физических адресов памяти.

Массивы данных и тегов в каждом банке являются независимыми. Эти четыре массива работают под общим управлением очереди формирования адресов памяти и схем внешнего интерфейса кристалла. В очереди адресов могут одновременно находиться до 16 команд загрузки и записи, которые обрабатываются в четырех отдельных конвейерах. Команды из этой очереди динамически подаются для выполнения в специальный конвейер, который обеспечивает вычисление исполнительного виртуального адреса и преобразование этого адреса в физический. Три других параллельно работающих конвейера могут одновременно выполнять проверку тегов, осуществлять пересылку данных для команд загрузки и завершать выполнение команд записи в память. Хотя команды выполняются в строгом порядке их расположения в памяти, вычисление адресов и пересылка данных для команд загрузки могут происходить неупорядоченно. Схемы внешнего интерфейса кристалла могут производить заполнение или обратное копирование строк кэш-памяти, либо операции просмотра тегов. Такая параллельная работа большинства устройств процессора позволяет процессору R12000 эффективно выполнять реальные многопроцессорные приложения.

Работа конвейеров кэш-памяти данных тесно координирована. Например, команды загрузки могут выполнять проверку тегов и чтение данных в том же такте, что и преобразование адреса. Команды записи сразу же начинают проверку тегов, чтобы, в случае необходимости, как можно раньше инициировать заполнение требуемой строки из L2, но непосредственная запись данных в кэш задерживается до тех пор, пока сама команда записи не станет са-

мой старой командой в общей очереди выполняемых команд и ей не будет позволено зафиксировать свой результат. Промах при обращении к L1 данных инициирует процесс заполнения строки из кэш-памяти второго уровня. При выполнении команд загрузки одновременно с заполнением строки кэш-памяти данные могут поступать по цепям обхода в регистровый файл.

При обнаружении промаха при обращении к кэш-памяти данных ее работа не блокируется, т.е. она может продолжать обслуживание следующих запросов. Это особенно полезно для уменьшения такого важного показателя качества реализованной архитектуры как среднее число тактов на команду (CPI - clock cycles per instruction). Эффект применения неблокируемой кэш-памяти сильно зависит от характеристик самих программ. Для небольших тестов, рабочие наборы которых полностью помещаются в L1, этот эффект не велик. Однако для более реальных программ выигрыш оказывается существенным.

Интерфейс L2 процессора R12000 поддерживает 128-разрядную магистраль данных, которая может работать с тактовой частотой до 300 МГц. Все стандартные синхронные сигналы управления статической памятью вырабатываются внутри процессора. Не требуется никаких внешних интерфейсных схем. Минимальный объем L2 составляет 512 Кбайт, максимальный размер - 16 Мбайт. Размер строки этой кэш-памяти программируется и может составлять 128 или 1024 байт.

Одним из методов улучшения временных показателей работы кэш-памяти является построение псевдо-множественно-ассоциативной кэш-памяти. В такой кэш-памяти частота промахов находится на уровне частоты промахов множественно-ассоциативной памяти, а время выборки при попадании соответствует кэш-памяти с прямым отображением.

Кэш-память R12000 организована именно таким способом, причем для ее реализации используются стандартные синхронные микросхемы памяти SRAM. В одном наборе микросхем памяти находятся оба канала кэша. Информация о частоте использования этих каналов хранится в схемах управления кэшем на процессорном кристалле. Поэтому после обнаружения промаха в первичном

кэше из наиболее часто используемого канала вторичного кэша считываются две четырехсловные строки.

Их теги считываются вместе с первой четырехсловной строкой, а теги альтернативного канала читаются одновременно со второй четырехсловной строкой - это осуществляется простым инвертированием старшего разряда адреса. При этом возможны три случая:

- если происходит попадание по первому каналу, то данные доступны немедленно,
- если попадание происходит по альтернативному каналу, выполняется повторное чтение вторичного КЭШа,
- если отсутствует попадание по обоим каналам, вторичный кэш должен перезаполняться из основной памяти.

Для обеспечения целостности данных в кэш-памяти большой емкости обычной практикой является использование кодов, исправляющих одиночные ошибки (ЕСС-кодов). В R12000 с каждой четырехсловной строкой хранится 9-разрядный ЕСС-код и бит четности. Дополнительный бит четности позволяет сократить задержку (поскольку проверка на четность может быть выполнена очень быстро), чтобы предотвратить использование некорректных данных. При этом, если обнаруживается корректируемая ошибка, то чтение повторяется через специальный двухтактный конвейер коррекции ошибок.

Объем внутренней двухканальной множественно-ассоциативной кэш-памяти команд составляет 32 Кбайт. В процессе ее загрузки команды частично декодируются. При этом к каждой команде добавляются 4 дополнительных бита, указывающих на исполнительное устройство, в котором будет выполняться команда. Таким образом, в кэш-памяти команды хранятся в 36-разрядном формате.

6.7. PowerPC 970 IBM

64-битные процессоры семейства PowerPC фирмы IBM находят широкое применение – от устройств для серверов старших моделей (Power4 и Power5) до встраиваемых процессоров (PowerPC 970 с высокопроизводительной векторной Altivec*

SIMD-поддержкой). Это совсем неплохо для архитектуры, совместимой с 32-бит процессорами семейства. Компания IBM пошла по пути "мультипроцессирования на уровне микросхемы" (Chip Level Multiprocessing – CMP). Нестандартная любопытная суперскалярная RISC-архитектура таких процессоров позволяет выдавать на выполнение до 8,5 команд/такт. Не усложняя конструкции процессора, на одном кристалле можно разместить два процессорных ядра с более чем десятком исполнительных устройств. Архитектура процессоров PowerPC оперирует более мощными и гибкими SIMD-командами, чем другие устройства, предназначенные для мультимедийной обработки. Altivec SIMD-поддержка обеспечивает еще большую универсальность, предоставляя возможность использовать 32 специализированных 128-бит векторных регистра, четыре регистровых операнда, 162 векторные команды, а также выполнение параллельных скалярных операций с плавающей точкой.

Процессор серии PowerPC 970, впервые представленный в 2002 году, достаточно дешевое 64-бит воплощение процессора Power4, предназначенное для серверов старших моделей. Хотя уменьшение размеров изделия редко позволяет рассчитывать на получение высоких характеристик, модернизация 64-бит микропроцессора Power4 и добавление векторной Altivec-поддержки позволили создать впечатляющий микропроцессор для серверов, графических рабочих станций, настольных компьютеров.

На кристалле Power4 площадью 415 мм², содержащем 170 млн. транзисторов, расположены два процессорных ядра на тактовую частоту 1,3 ГГц, L2 кэш емкостью 1,5 Мбит, контроллер L3 кэша и контроллер межчиповой связи, позволяющий размещать четыре Power4-чипа в одном 5184-выводном многокристальном модуле размером 85x85 мм. Четыре таких модуля могут быть объединены без связующих логических схем. Получаемая в результате 32-канальная микропроцессорная подсистема имеет более 20 тыс. контактных площадок ввода-вывода и рассеивает ~2 кВт.

Разработчики IBM оставили одно процессорное ядро процессора Power4, удалив контроллер L3 кэш, сложный межчиповый контроллер и добавив Altivec-расширение. Процессор PowerPC 970, выполненный по 0,13-мкм КМОП-технологии на КНИ-

подложке с восьмислойной медной металлизацией, содержит 52 млн. транзисторов и размещается на кристалле площадью 118 мм². Он имеет два блока L1 с контролем по четности (команд и данных, емкостью 64К и 32 Кбит, соответственно) и L2 с корректировкой ошибок емкостью 512 Кбит. Монтируется он в 576-выводной керамический BGA-корпус размером 25x25 мм.

В результате разработчики получили высокопроизводительный, более традиционный 64-битный микропроцессор с одним ядром и малой потребляемой мощностью (19 Вт при напряжении питания 1,1 В и тактовой частоте 1,2 ГГц и 42 Вт при напряжении 1,3 В и частоте 1,8 ГГц), не требующий сборки в дорогой многокристальный модуль. Наряду с этим, у него более глубокий конвейер (16 ступеней), предусмотрены динамическое предсказание переходов, шина с высокой пропускной способностью (до 7,1 Гбайт/с) и достаточное логическое обрамление, что делает его предпочтительным при выборе процессора SMP-системы.

С 2004 года IBM объявила о массовом производстве процессора PowerPC 970FX по 90-нм КНИ-технологии в сочетании с технологиями напряженного кремния и восьмислойных медных соединений. В микросхеме использован новый метод сложной настройки и управления тактовой частотой и пороговым напряжением транзисторов, позволяющий регулировать эти параметры с шагом 0,5 МГц и 0,5 мВ, соответственно. В результате можно получать четыре различных значения порогового напряжения транзисторов с тонким затворным окислом и еще два значения напряжения для транзисторов с толстым окисным слоем. Это привело к снижению потребляемой мощности процессора до 15 Вт.

PowerPC 970FX находит самое широкое применение – от настольных компьютеров до серверов, накопителей и связанных систем. Компания Apple уже сообщила о намерении использовать его в новом стоечном сервере Xserve G5 1U. Микропроцессор PowerPC 970FX уже получил премию аналитиков Microprocessor Report как лучший процессор настольных машин, обойдя Pentium 4 фирмы Intel и Athlon 64 FX-51 компании AMD.

7. Микропроцессоры нетрадиционных архитектур

7.1. Ассоциативные процессоры

Существующие в настоящее время алгоритмы прикладных задач, системное программное обеспечение и аппаратные средства преимущественно ориентированы на традиционную адресную обработку данных. Данные должны быть представлены в виде ограниченного количества форматов (например, массивы, списки, записи), должна быть явно создана структура связей между элементами данных посредством указателей на адреса элементов памяти, при обработке этих данных должна быть выполнена совокупность операций, обеспечивающих доступ к данным по указателям. Такой подход обуславливает громоздкость операционных систем и систем программирования, а также служит препятствием к созданию вычислительных средств с архитектурой, ориентированной на более эффективное использование параллелизма обработки данных.

Ассоциативный способ обработки данных позволяет преодолеть многие ограничения, присущие адресному доступу к памяти, за счет задания некоторого критерия отбора и проведение требуемых преобразований, только над теми данными, которые удовлетворяют этому критерию. Критерием отбора может быть совпадение с любым элементом данных, достаточным для выделения искомым данных из всех данных. Поиск данных может происходить по фрагменту, имеющему большую или меньшую корреляцию с заданным элементом данных.

Исследованы и в разной степени используются несколько подходов, различающихся полнотой реализации модели ассоциативной обработки. Если реализуется только ассоциативная выборка данных с последующим поочередным использованием найденных данных, то говорят об ассоциативной памяти или памяти, адресуемой по содержимому. При достаточно полной реализации всех свойств ассоциативной обработки, используется термин «ассоциативный процессор».

Ассоциативные системы относятся к классу: один поток команд - множество потоков данных (SIMD = Single Instruction Multiple Data). Эти системы включают большое число операцион-

ных устройств, способных одновременно по командам управляющего устройства вести обработку нескольких потоков данных. В ассоциативных вычислительных системах информация на обработку поступает от ассоциативных запоминающих устройств (АЗУ), характеризующиеся тем, что информация в них выбирается не по определенному адресу, а по ее содержанию.

7.2. Матричные процессоры

Наиболее распространенными из систем, класса: один поток команд - множество - потоков данных (SIMD), являются матричные системы, которые лучше всего приспособлены для решения задач, характеризующихся параллелизмом независимых объектов или данных. Организация систем подобного типа на первый взгляд достаточно проста. Они имеют общее управляющее устройство, генерирующее поток команд и большое число процессорных элементов, работающих параллельно и обрабатывающих каждая свой поток данных.

Таким образом, производительность системы оказывается равной сумме производительностей всех процессорных элементов. Однако на практике, чтобы обеспечить достаточную эффективность системы при решении широкого круга задач необходимо организовать связи между процессорными элементами с тем, чтобы наиболее полно загрузить их работой. Именно характер связей между процессорными элементами и определяет разные свойства системы.

Одним из первых матричных процессоров был SOLOMON (60-е годы) (рис. 24). Система SOLOMON содержит 1024 процессорных элемента, соединены в виде матрицы: 32×32 . Каждый процессорный элемент матрицы включает в себя процессор, обеспечивающий выполнение последовательных поразрядных арифметических и логических операций, а также оперативное ЗУ, емкостью 16 Кбайт. Длина слова - переменная от 1 до 128 разрядов. Разрядность слов устанавливается программно. По каналам связи от устройства управления передаются команды и общие константы.

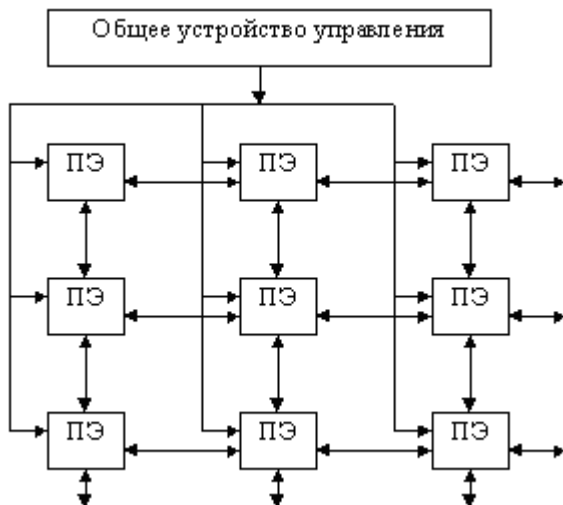


Рис. 24

В процессорном элементе используется, так называемая, многомодальная логика, которая позволяет каждому процессорному элементу выполнять или не выполнять общую операцию в зависимости от значений обрабатываемых данных. В каждый момент все активные процессорные элементы выполняют одну и ту же операцию над данными, хранящимися в собственной памяти и имеющими один и тот же адрес.

Идея многомодальности заключается в том, что в каждом процессорном элементе имеется специальный регистр на 4 состояния - регистр моды. Мода (модальность) заносится в этот регистр от устройства управления. При выполнении последовательности команд модальность передается в коде операции и сравнивается с содержимым регистра моды. Если есть совпадения, то операция выполняется. В других случаях процессорный элемент не выполняет операцию, но может, в зависимости от кода, пересылать свои операнды соседнему процессорному элементу. Такой механизм позволяет выделить строку или столбец процессорных элементов, что очень полезно при операциях над матрицами. Взаимодействуют процессорные элементы с периферийным оборудованием через внешний процессор.

Дальнейшим развитием матричных процессоров стала система ILLIAS-4, разработанная фирмой Varoys. Первоначально система должна была включать в себя 256 процессорных элементов, разбитых на группы, каждый из которых должен управляться специальным процессором. Однако по различным причинам была создана система, содержащая одну группу процессорных элементов и управляющий процессор. Если в начале предполагалось достичь быстродействия порядка 1 млрд. операций в секунду, то реальная система работала с быстродействием около 200 млн. операций в секунду. Эта система в течение ряда лет считалась одной из самых высокопроизводительных в мире.

7.3. ДНК процессоры

В настоящее время в поисках реальной альтернативы полупроводниковым технологиям создания новых вычислительных систем ученые обращают все большее внимание на биотехнологии, или биокомпьютинг, который представляет собой гибрид информационных, молекулярных технологий, также биохимии. Биокомпьютинг позволяет решать сложные вычислительные задачи, пользуясь методами, принятыми в биохимии и молекулярной биологии, организуя вычисления при помощи живых тканей, клеток, вирусов и биомолекул.

Наибольшее распространение получил подход, где в качестве основного элемента (процессора) используются молекулы дезоксирибонуклеиновой кислоты. Центральное место в этом подходе занимает так называемый ДНК - процессор. Кроме ДНК в качестве био-процессора могут быть использованы также белковые молекулы и биологические мембраны.



Так же, как и любой другой процессор, ДНК процессор характеризуется структурой и набором команд. В нашем случае структура процессора - это структура молекулы ДНК. А набор команд - это перечень биохимических операций с молекулами.

кулами.

Принцип устройства компьютерной ДНК-памяти основан на последовательном соединении четырех нуклеотидов (основных кирпичиков ДНК-цепи). Три нуклеотида, соединяясь в любой последовательности, образуют элементарную ячейку памяти - кодон, которые затем формируют цепь ДНК. Основная трудность в разработке ДНК-компьютеров связана с проведением избирательных однокодонных реакций (взаимодействий) внутри цепи ДНК. Однако прогресс есть уже и в этом направлении. Уже есть экспериментальное оборудование, позволяющее работать с одним из 1020 кодонов или молекул ДНК. Другой проблемой является самосборка ДНК, приводящая к потере информации. Ее преодолевают введением в клетку специальных ингибиторов - веществ, предотвращающих химическую реакцию самосшивки.

Использование молекул DNA для организации вычислений – это не слишком новая идея. Теоретическое обоснование подобной возможности было сделано еще в 50-х годах прошлого века (Р.П. Фейманом). В деталях эта теория была проработана в 70-х годах Ч. Бенеттом и в 80-х М. Конрадом.

Первый компьютер на базе ДНК был создан еще в 1994 г. американским ученым Леонардом Адлеманом. Он смешал в пробирке молекулу ДНК, в которой были закодированы исходные данные, и специальным образом подобранные ферменты. В результате химической реакции структура ДНК изменилась таким образом, что в ней в закодированном виде был представлен ответ задачи. Поскольку вычисления проводились в ходе химической реакции с участием ферментов, на них было затрачено очень мало времени.

Ричард Липтон из Принстона первым показал, как, используя ДНК, кодировать двоичные числа и решать проблему удовлетворения логического выражения. Суть ее в том, что, имея некоторое логическое выражение, включающее n логических переменных, нужно найти все комбинации значений переменных, делающих выражение истинным. Задачу можно решить только перебором 2^n комбинаций. Все эти комбинации легко закодировать с помощью ДНК, а дальше действовать по методике Адлемана.

Первую модель биокомпьютера, правда, в виде механизма из пластмассы, в 1999 г. создал Ихуд Шапиро из Вейцмановского

института естественных наук. Она имитировала работу “молекулярной машины” в живой клетке, собирающей белковые молекулы по информации с ДНК, используя РНК в качестве посредника между ДНК и белком.

А в 2001 г. Шапиро удалось реализовать вычислительное устройство на основе ДНК, которое может работать почти без вмешательства человека. Система имитирует машину Тьюринга — одну из фундаментальных концепций вычислительной техники. Машина Тьюринга шаг за шагом считывает данные и в зависимости от их значений принимает решения о дальнейших действиях. Теоретически она может решить любую вычислительную задачу. По своей природе молекулы ДНК работают аналогичным образом, распадаясь и рекомбинируя в соответствии с информацией, закодированной в цепочках химических соединений.

Разработанная в Вейцмановском институте установка кодирует входные данные и программы в состоящих из двух цепей молекулах ДНК и смешивает их с двумя ферментами. Молекулы фермента выполняли роль аппаратного, а молекулы ДНК - программного обеспечения. Один фермент расщепляет молекулу ДНК с входными данными на отрезки разной длины в зависимости от содержащегося в ней кода. А другой рекомбинирует эти отрезки в соответствии с их кодом и кодом молекулы ДНК с программой. Процесс продолжается вдоль входной цепи, и, когда доходит до конца, получается выходная молекула, соответствующая конечному состоянию системы.

Этот механизм может использоваться для решения самых разных задач. Хотя на уровне отдельных молекул обработка ДНК происходит медленно - с типичной скоростью от 500 до 1000 бит/с, что во много миллионов раз медленнее современных кремниевых процессоров, по своей природе она допускает массовый параллелизм. По оценкам Шапиро и его коллег, в одной пробирке может одновременно происходить триллион процессов, так что при потребляемой мощности в единицы нановатт может выполняться миллиард операций в секунду.

В 2002 г. фирма Olympus Optical разработала версию ДНК-компьютера, предназначенного для генетического анализа. Он имеет молекулярную и электронную составляющие. Первая осу-

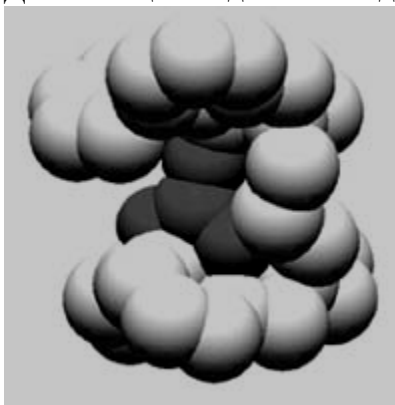
ществляет химические реакции между молекулами ДНК, обеспечивает поиск и выделение результата вычислений. Вторая - обрабатывает информацию и анализирует полученные результаты.

Возможностями биокомпьютеров заинтересовались и военные. Американское агентство по исследованиям в области обороны DARPA выполняет проект, получивший название Bio-Comp (Biological Computations, биологические вычисления). Его цель - создание мощных вычислительных систем на основе ДНК.

Пока до практического применения компьютеров на базе ДНК еще очень далеко. Однако в будущем их смогут использовать не только для вычислений, но и как своеобразные нанофабрики лекарств. Поместив подобное "устройство" в клетку, врачи смогут влиять на ее состояние, исцеляя человека от самых опасных недугов.

7.4. Клеточные процессоры

Клеточные процессоры представляют собой самоорганизующиеся колонии различных "умных" микроорганизмов, в геном которых удалось включить некую логическую схему, которая могла бы активизироваться в присутствии определенного вещества. Для этой цели идеально подошли бы бактерии, стакан с которыми и представлял бы собой компьютер. Такие компьютеры очень дешевы в производстве. Им не нужна столь стерильная атмосфера, как при производстве полупроводников.



Главным свойством процессора такого рода является то, что каждая их клетка представляет собой миниатюрную химическую лабораторию. Если биоорганизм запрограммирован, то он просто производит нужные вещества. Достаточно вырастить одну клетку, обладающую заданными качествами, и можно легко и быстро вырастить тысячи клеток с такой же программой.

Достаточно вырастить одну клетку, обладающую заданными качествами, и можно легко и быстро вырастить тысячи клеток с такой же программой.

Основная проблема, с которой сталкиваются создатели клеточных биокомпьютеров, - организация всех клеток в единую работающую систему. На сегодняшний день практические достижения в области клеточных компьютеров напоминают достижения 20-х годов в области ламповых и полупроводниковых компьютеров. В Лаборатории искусственного интеллекта Массачусетского технологического университета создана клетка, способная хранить на генетическом уровне 1 бит информации. Также разрабатываются технологии, позволяющие единичной бактерии отыскивать своих соседей, образовывать с ними упорядоченную структуру и осуществлять массив параллельных операций.

В 2001 г. американские ученые создали трансгенные микроорганизмы (т. е. микроорганизмы с искусственно измененными генами), клетки которых могут выполнять логические операции И и ИЛИ.

Специалисты лаборатории Оук-Ридж, штат Теннесси, использовали способность генов синтезировать тот или иной белок под воздействием определенной группы химических раздражителей. Ученые изменили генетический код бактерий *Pseudomonas putida* таким образом, что их клетки обрели способность выполнять простые логические операции. Например, при выполнении операции И в клетку подаются два вещества (по сути - входные операнды), под влиянием которых ген вырабатывает определенный белок. Теперь учеными ведутся работы по созданию на базе этих клеток более сложных логических элементов, а также работы по созданию клетки, выполняющей параллельно несколько логических операций.

Потенциал биокомпьютеров очень велик. К достоинствам, выгодно отличающим их от компьютеров, основанных на кремниевых технологиях, относятся:

- 1) более простая технология изготовления, не требующая для своей реализации столь жестких условий, как при производстве полупроводников

- 2) использование не бинарного, а тернарного кода (информация кодируется тройками нуклеотидов), что позволит при меньшем количестве шагов перебрать большее число вариантов при анализе сложных систем

3) потенциально исключительно высокая производительность, которая может составлять до 10^{14} операций в секунду за счет одновременного вступления в реакцию триллионов молекул ДНК

4) возможность хранить данные с плотностью, в триллионы раз превышающей показатели оптических дисков

5) исключительно низкое энергопотребление

Однако, наряду с очевидными достоинствами, биокомпьютеры имеют и существенные недостатки, такие как:

1) сложность со считыванием результатов - современные способы определения кодирующей последовательности не совершенны, сложны, трудоемки и дороги

2) низкая точность вычислений, связанная с возникновением мутаций, прилипанием молекул к стенкам сосудов и т.д.

3) невозможность длительного хранения результатов вычислений в связи с распадом ДНК в течение времени

Хотя до практического использования биокомпьютеров еще очень далеко, но предполагается, что, они найдут достойное применение в медицине и фармакологии, а также с их помощью станет возможным объединение информационных и биотехнологий.

7.5. Коммуникационные процессоры

Коммуникационные процессоры - это микрочипы, являющие собой нечто среднее между жесткими специализированными интегральными микросхемами и гибкими процессорами общего назначения.

Коммуникационные процессоры программируются, как и обычные процессоры, но построены с учетом сетевых задач, оптимизированы для сетевой работы, и на их основе производители - как процессоров, так и оборудования - пишут программное обеспечение для специфических приложений.

Коммуникационный процессор имеет собственную память и оснащен высокоскоростными внешними каналами для соединения с другими процессорными узлами. Его присутствие позволяет в значительной мере освободить вычислительный процессор от нагрузки, связанной с передачей сообщений между процессорными

ми узлами. Скоростной коммуникационный процессор с RISC-ядром позволяет управлять обменом данными по нескольким независимым каналам, поддерживать практически все распространенные протоколы обмена, гибко и эффективно распределять и обрабатывать последовательные потоки данных с временным разделением каналов.

Сама идея создания процессоров, предназначенных для оптимизации сетевой работы - и при этом достаточно универсальных для программной модификации – родилась в связи с необходимостью устранить различия в подходах к созданию локальных сетей (различные подходы к архитектуре сети, классификации потоков, и т.д

Новая серия коммуникационных процессоров Intel IXP4xx построена на базе распределенной архитектуры XScale и включает мощные мультимедийные возможности, а также развитые сетевые интерфейсы Ethernet. Сочетание высокой производительности и низкого энергопотребления позволяет эффективно применять коммуникационные процессоры Intel не только в классических сетевых приложениях, но и для построения интернет-ориентированных встраиваемых систем промышленного назначения.

Эффективность работы промышленных предприятий сегодня напрямую зависит от гибкости применяемых систем автоматизированного управления. Крупные производственные установки требуют использования нескольких децентрализованных систем управления, связанных друг с другом мощной информационной сетью, способной работать в сложных промышленных условиях. Зачастую эти средства промышленной коммуникации призваны обеспечить возможность гибкого управления, программирования и контроля работы распределенных систем управления из удаленных диспетчерских пунктов. Осуществление этих целей возможно с помощью коммуникационных процессоров, предназначенных для подключения персональных компьютеров к промышленным информационным сетям.

Дополнительные возможности, обеспечиваемые коммуникационными процессорами должны быть интересны, прежде всего, тем пользователям, которым необходимо осуществлять сложные

транзакции или наладить прямую голосовую и видео передачи в рамках сетевой инфраструктуры.

7.6. Процессоры баз данных

Процессорами (машинами) баз данных в настоящее время принято называть программно- аппаратные комплексы, предназначенные для выполнения всех или некоторых функций систем управления базами данных (СУБД). Если в свое время системы управления базами данных предназначались в основном для хранения текстовой и числовой информации, то теперь они рассчитаны на самые различные форматы данных, в том числе графические, звуковые и видео. Процессоры баз данных выполняют функции управления и распространения, обеспечивают дистанционный доступ к информации через шлюзы, а также репликацию обновленных данных с помощью различных механизмов тиражирования.

Современные процессоры баз данных должны обеспечивать естественную связь накапливаемой в базах данных информации со средствами оперативной обработки транзакций и Internet-приложениями. Это должны быть системы, которые дают пользователям возможность в любой момент обратиться к корпоративным данным и проанализировать их, вне зависимости от того, где эти данные размещаются.

Решение таких задач требует существенного увеличения производительности таких систем. Однако традиционная программная реализация многочисленных функций современных СУБД на ЭВМ общего назначения приводит к громоздким и непроизводительным системам с недостаточно высокой надежностью. Необходим поиск новых архитектурных и аппаратных решений. Интенсивные исследования, проводимые в этой области в настоящее время, привели к пониманию необходимости использования в качестве процессоров баз данных специализированных параллельных вычислительных систем. Создание такого рода систем связывается с реализацией параллелизма при выполнении последовательности операций и транзакций, а также конвейерной потоковой обработки данных.

7.7. Поточковые процессоры

Потоковыми называют процессора, в основе работы которых лежит принцип обработки многих данных с помощью одной команды. Согласно классификации Флинна они принадлежат к SIMD архитектуре. Технология SIMD позволяет выполнять одно и то же действие, например вычитание и сложение, над несколькими наборами чисел одновременно. SIMD-операции для чисел двойной точности с плавающей запятой ускоряют работу ресурсоемких приложений для создания контента, трехмерного рендеринга, финансовых расчетов и научных задач. Кроме того, усовершенствованы возможности 64-разрядной технологии MMX (целочисленных SIMD-команд); эта технология распространена на 128-разрядные числа, что позволяет ускорить обработку видео, речи, шифрование, обработку изображений и фотографий. Поточковый процессор повышает общую производительность, что особенно важно при работе с 3D-графическими объектами.

Может быть отдельный потоковый процессор (Single-streaming processor — SSP) и многопоточковый процессор (Multi-Streaming Processor - MSP).

Ярким представителем потоковых процессоров является семейство процессоров Intel, начиная с Pentium III, в основе работы которых лежит технология Streaming SIMD Extensions (SSE, потоковая обработка по принципу "одна команда - много данных"). Эта технология позволяет выполнять такие сложные и необходимые в век Internet задачи, как обработка речи, кодирование и декодирование видео- и аудиоданных, разработка трехмерной графики и обработка изображений.

Беспорными представителями класса SIMD считаются матрицы процессоров: ILLIAC IV, ICL DAP, Goodyear Aerospace MPP, Connection Machine 1 и т.п. В таких системах единое управляющее устройство контролирует множество процессорных элементов. Каждый процессорный элемент получает от устройства управления в каждый фиксированный момент времени одинаковую команду и выполняет ее над своими локальными данными.

Другими представителями SIMD-класса являются векторные процессоры, в основе которых лежит векторная обработка

данных. Векторная обработка увеличивает производительность процессора за счет того, что обработка целого набора данных (вектора) производится одной командой. Векторные компьютеры манипулируют массивами сходных данных подобно тому, как скалярные машины обрабатывают отдельные элементы таких массивов. В этом случае каждый элемент вектора надо рассматривать как отдельный элемент потока данных. При работе в векторном режиме векторные процессоры обрабатывают данные практически параллельно, что делает их в несколько раз более быстрыми, чем при работе в скалярном режиме. Максимальная скорость передача данных в векторном формате может составлять 64 Гб/с, что на 2 порядка быстрее, чем в скалярных машинах. Примерами систем подобного типа является, например, процессоры фирм NEC и Hitachi.

7.8. Процессоры с многозначной (нечеткой) логикой

Идея построения процессоров с нечеткой логикой (fuzzy logic) основывается на нечеткой математике. Математическая теория нечетких множеств, предложенная проф. Л.А. Заде, являясь предметом интенсивных исследований, открывает все большие возможности перед системными аналитиками. Основанные на этой теории различные компьютерные системы, в свою очередь, существенно расширяют область применения нечеткой логики.

Подходы нечёткой математики дают возможность оперировать входными данными, непрерывно меняющимися во времени и значениями, которые невозможно задать однозначно, такими, например, как результаты статистических опросов. В отличие от традиционной формальной логики, известной со времен Аристотеля и оперирующей точными и четкими понятиями типа истина и ложь, да и нет, ноль и единица, нечеткая логика имеет дело со значениями, лежащими в некотором (непрерывном или дискретном) диапазоне.

Функция принадлежности элементов к заданному множеству также представляет собой не жесткий порог "принадлежит - не принадлежит", а плавную сигмоиду, проходящую все значения от нуля до единицы. Теория нечеткой логики позволяет выполнять

над такими величинами весь спектр логических операций - объединение, пересечение, отрицание и др.

Согласно знаменитой теореме FAT (Fuzzy Approximation Theorem), доказанной Коско, любая математическая система может быть аппроксимирована системой, основанной на нечеткой логике. Свое второе рождение теория нечеткой логики пережила в начале восьмидесятых годов, когда сразу несколько групп исследователей (в основном в США и Японии) всерьез занялись созданием электронных систем различного применения, использующих нечеткие управляющие алгоритмы. Используя преимущества нечеткой логики, заключающиеся в простоте содержательного представления, можно упростить проблему, представить ее в более доступном виде и повысить производительность системы.

Задачи с помощью нечеткой логики решаются по следующему принципу:

- 1) численные данные (показания измерительных приборов, результаты анкетирования) фаззируются (переводятся в нечеткий формат);
- 2) обрабатываются по определенным правилам;
- 3) дефаззируются и в виде привычной информации подаются на выход.

Оказалось возможным создание нечеткого процессора, позволяющего выполнять различные нечеткие операции и приближенные рассуждения (нечеткий вывод) в соответствии с правилами логического вывода. В 1986 году в AT&T Bell Labs создавались процессоры с “прошитой” нечеткой логикой обработки информации.

В начале 90-х компания Adaptive Logic из США выпустила кристалл, сделанный по аналогово-цифровой технологии (рис. 25). Он позволит сократить сроки конструирования многих встроенных систем управления реального времени, заменив собой традиционные схемы нечетких микроконтроллеров. Аппаратный процессор нечеткой логики второго поколения принимает аналоговые сигналы, переводит их в нечеткий формат, затем, применяя соответствующие правила, преобразует результаты в формат обычной логики и далее – в аналоговый сигнал.

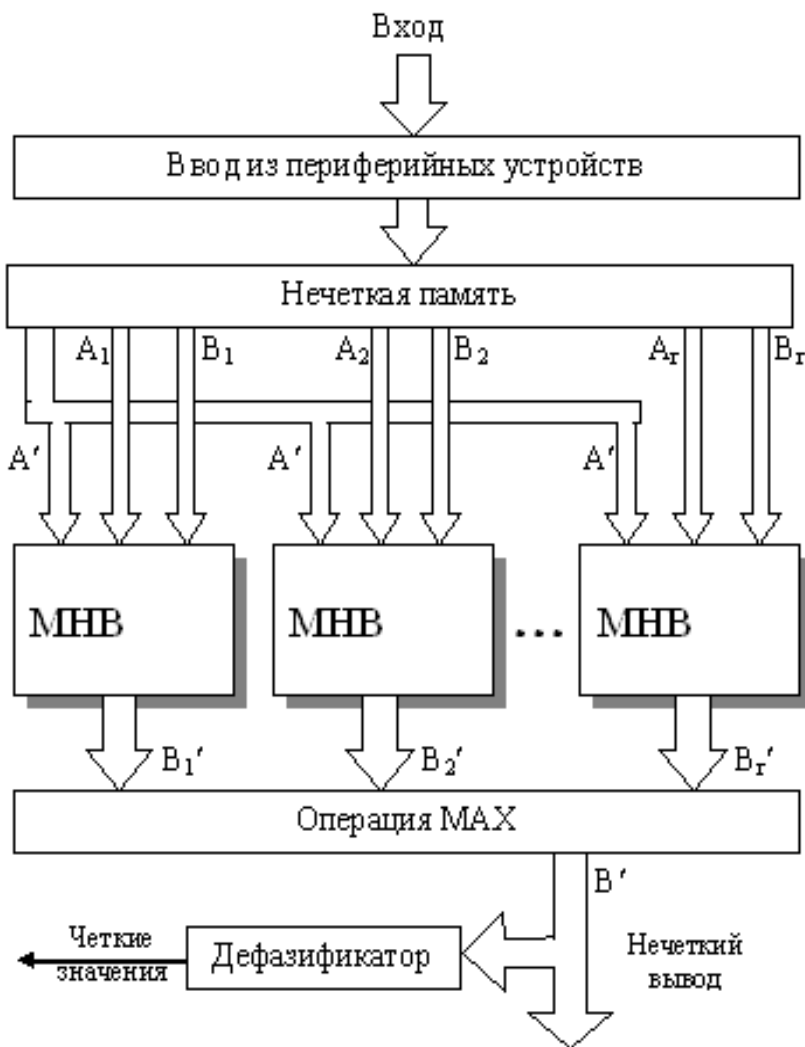


Рис. 25

Все это осуществляется без внешних запоминающих устройств, преобразователей и какого бы ни было программного обеспечения нечеткой логики. Этот микропроцессор относительно прост по сравнению с громоздкими программными обеспечениями. Но так как его основу составляет комбинированный цифровой/аналоговый кристалл, он функционирует на очень высоких скоро-

стях (частота отсчетов входного сигнала – 10 кГц, а скорость расчета – 500 тыс. правил/с), что во многих случаях приводит к лучшим результатам в системах управления по сравнению с более сложными, но медлительными программами.

В Европе и США ведутся интенсивные работы по интеграции fuzzy команд в ассемблеры промышленных контроллеров встроенных устройств (чипы Motorola 68HC11. 12. 21). Такие аппаратные средства позволяют в несколько раз увеличить скорость выполнения приложений и компактность кода по сравнению с реализацией на обычном ядре. Кроме того, разрабатываются различные варианты fuzzy- сопроцессоров, которые контактируют с центральный процессор через общую шину данных, концентрируют свои усилия на размывании/ уплотнении информации и оптимизации использования правил (продукты Siemens Nixdorf).

Идеи нечеткой логики не являются панацеей и не смогут совершить переворот в компьютерном мире. Нечеткая логика не решит тех задач, которые не решаются на основе логики двоичной, но во многих случаях она удобнее, производительнее и дешевле. Разработанные на ее основе специализированные аппаратные решения (fuzzy-вычислители) позволят получить реальные преимущества в быстродействии. Если каскадировать fuzzy-вычислители, мы получим один из вариантов нейропроцессора или нейронной сети. Во многих случаях эти понятия просто объединяют, называя общим термином «neuro-fuzzy logic».

В настоящее время перспективой использовать процессоры, основанные на нечеткой логике всерьез заинтересовались военные. Известно, что NASA рассматривает возможность применения (если еще не применяет) нечеткие системы для управления процессами стыковки космических аппаратов.

7.9. Сигнальные процессоры

В ответ на возросшие запросы потребителей фирма Motorola разработала новую архитектуру микросхемы, ориентированную как на выполнение сложных алгоритмов цифровой обработки сигналов, так и на решение задач управления. Семейство микросхем DSP568xx построено на базе ядра 16-разрядного про-

цессора DSP56800 с фиксированной точкой. Это ядро предназначено для эффективного решения задач управления и цифровой обработки сигналов. Реализованный в нем набор команд обеспечивает цифровую обработку сигналов с эффективностью лучших DSP общего назначения и отвечает требованиям простоты создания компактных программ управления.

Ядро DSP56800 является программируемым 16-разрядным КМОП процессором, предназначенным для выполнения цифровой обработки сигналов в реальном масштабе времени и решения вычислительных задач. Ядро DSP56800 (рис. 26) состоит из четырех функциональных устройств: управления программой, генерации адресов, арифметико-логической обработки данных, обработки битов. Для увеличения производительности операции в устройствах выполняются параллельно. Каждое из устройств имеет свой набор регистров и логику управления и организовано таким образом, что может функционировать независимо и одновременно с тремя другими. Внутренние шины адресов и данных связывают между собой память, функциональные и периферийные устройства (регистры периферийных устройств расположены в области памяти). Таким образом, ядро реализует одновременное выполнение нескольких действий: устройство управления выбирает первую команду, устройство генерации адресов формирует до двух адресов второй команды, а АЛУ выполняет умножение третьей команды. Есть альтернативная возможность: в третьей команде операцию может выполнять не АЛУ, а устройство обработки битов. Конвейерная архитектура позволяет реализовать параллельную работу устройств, входящих в состав микросхемы, и существенно сократить время выполнения программы.

Конвейерная архитектура ядра DSP56800 оптимизирована для обеспечения эффективности цифровой обработки сигналов, компактности программ управления и обработки сигналов, и удобства программирования. Ниже приведены некоторые характеристики сигнального процессора:

- производительность 40 MIPS при тактовой частоте 80 МГц и напряжении питания 2.7...3.6 В,

- наличие набора команд совмещенной обработки, имеющих режимы адресации, характерные для программ цифровой обработки сигналов,

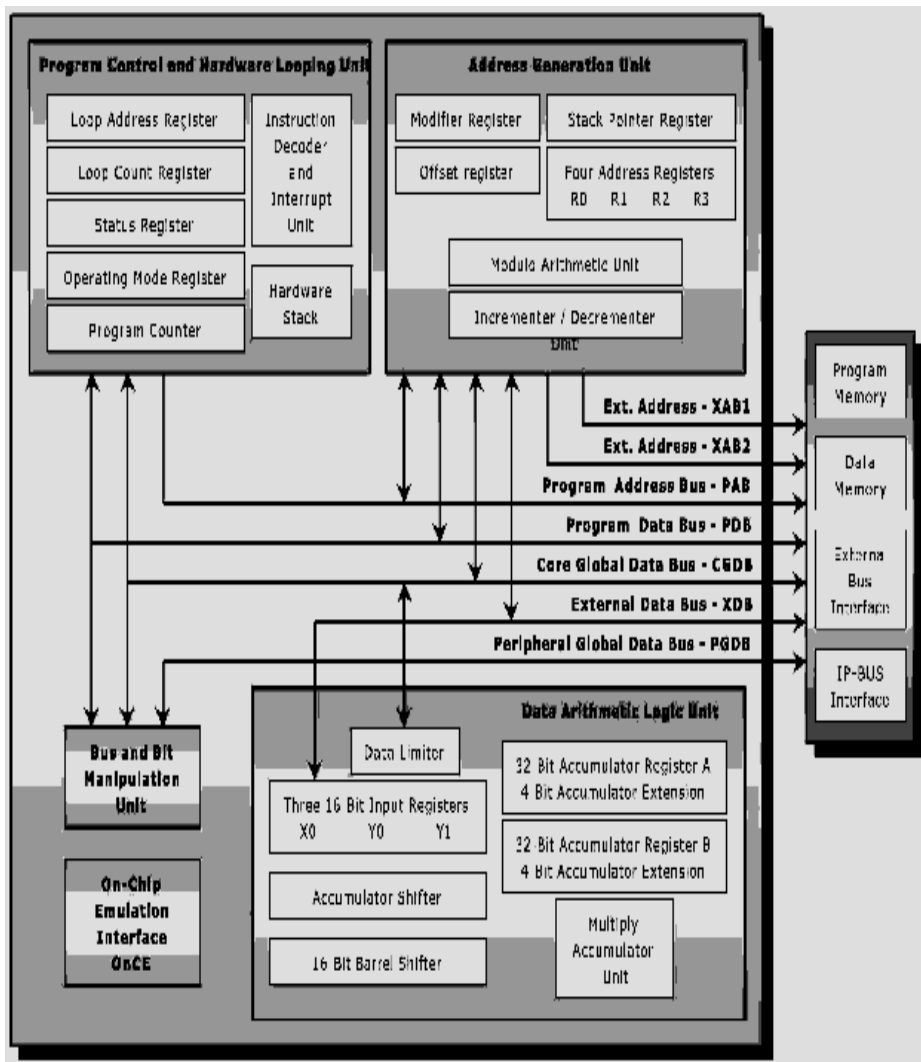


Рис. 26

- одноктактный параллельный 16x16 умножитель-сумматор,

- два 36-разрядных аккумулятора, включая биты расширения,
- однократное 16-разрядное устройство циклического сдвига,
- аппаратная реализация команд DO и REP,
- три внутренние 16-разрядные шины данных и три 16-разрядные шины адреса,
- одна 16-разрядная шина внешнего интерфейса,
- набор команд управления и цифровой обработки,
- режимы адресации такие же, как в сигнальных процессорах, и команды, снижающие объем программы,
- эффективный компилятор языка C и поддержка локальных переменных,
- стек подпрограмм и прерываний, не имеющий ограничения по глубине.

Для любого высокопроизводительного вычислителя, например цифрового сигнального процессора, критичным является процесс ввода/вывода данных с большой скоростью, т. к. при этом замедляется обработка данных. Снижения производительности можно избежать путем использования гибкого набора команд совмещенной с выполнением вычислительных операций передачи данных. Реализованы два типа операций совмещенной передачи - одинарная совмещенная передача и вдвоенное совмещенное чтение. Оба типа операций существенно повышают скорость цифровой обработки сигналов и численных расчетов. Все команды DSP56800 с совмещенной передачей выполняются за один командный цикл и занимают одно слово в памяти программ.

Однократная совмещенная передача позволяет выполнить арифметическое действие и одну передачу данных (чтение или запись) за один командный цикл. Например, можно одной командой выполнить сложение двух чисел и одновременно данные из регистра АЛУ записать в память. Одновременно с этим в устройстве вычисления адресов производятся соответствующие вычисления.

Команда типа двойного совмещенного чтения допускает выполнение арифметической операции и чтения двух величин из X-памяти данных в одной команде за один командный цикл. Например, можно в одной команде выполнить умножение двух чи-

сел, просуммировать с третьим, округлить результат и одновременно выполнить пересылку двух чисел из X-памяти данных в два регистра АЛУ.

Обычные микроконтроллеры, как правило, имеют объем встроенной в микросхему памяти, достаточный для выполнения сложных алгоритмов управления без использования дополнительной внешней памяти. Многие микросхемы DSP содержат встроенную память небольшого объема и, как правило, им требуется внешняя память для хранения программы. Микросхемы же семейства DSP56F8xx имеют встроенную память большого объема. Гарвардская архитектура DSP обеспечивает наличие двух независимых областей памяти - данных и программ. Для хранения в микросхеме данных и программ используется встроенная оперативная память и флэш-память.

Объем памяти каждого типа для микросхем семейства DSP56F8xx приведен в табл. 5. Как память программ, так и память данных могут быть расширены путем подключения внешней памяти. Микросхемы DSP56F803, DSP56F805, DSP56F807 допускают расширение объема внешней памяти до 64 К слов.

Таблица 5

Встроенная память	DSP56F801	DSP56F803	DSP56F805	DSP56F807
Флэш-память программы	8k x 16	32k x 16	32k x 16	60k x 16
Флэш-память данных	2k x 16	4k x 16	4k x 16	8k x 16
ОЗУ программ	1k x 16	512 x 16	512 x 16	2k x 16
ОЗУ данных	1k x 16	2k x 16	2k x 16	4k x 16
Флэш-память программы загрузки	2k x 16	2k x 16	2k x 16	2k x 16

Широкий набор периферийных устройств обычно являлся основной характеристикой микроконтроллеров, встраиваемых в устройства общего назначения. С другой стороны, обычные DSP были ориентированы на численную обработку сигналов и не содержали полного набора встроенных периферийных устройств, необходимых для решения задач управления. Использование внешних периферийных устройств приводит к увеличению числа

микросхем, усложнению платы и существенному возрастанию стоимости изделия.

Микросхемы семейств DSP56F8xx (рис. 27, табл. 6) имеют широкий набор встроенных периферийных устройств, пригодных для использования в системах управления всех типов

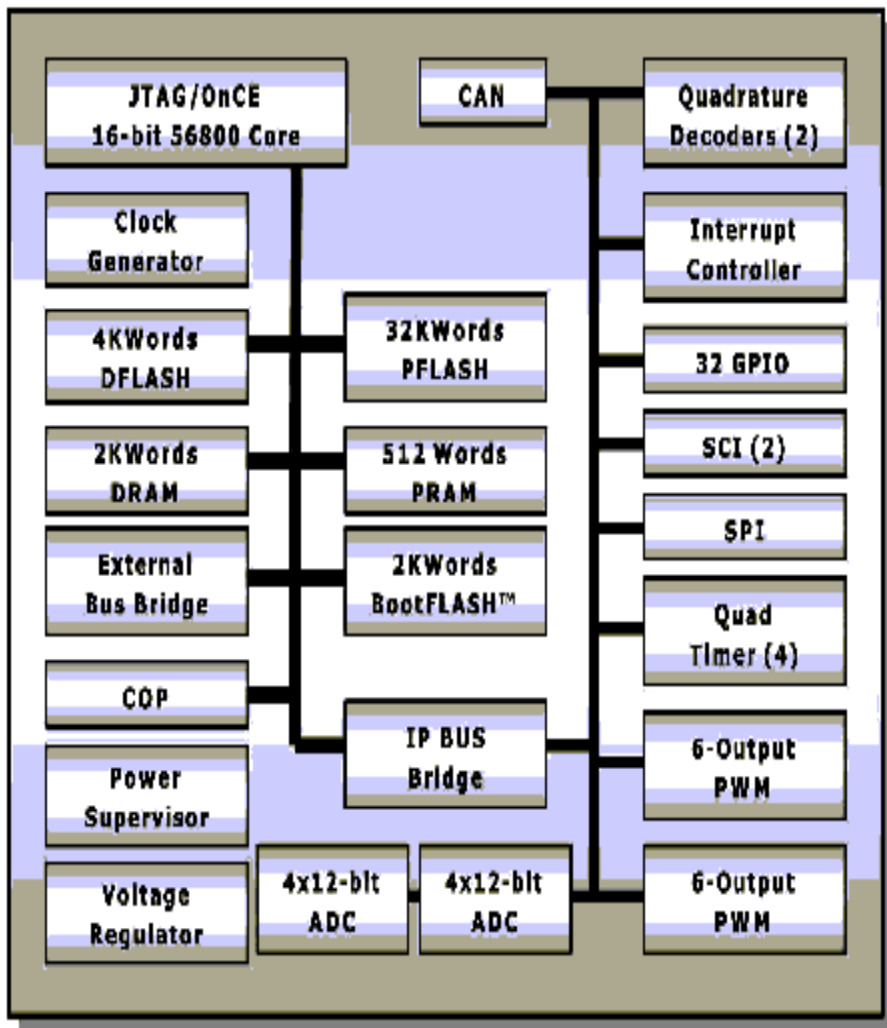


Рис. 27

Таблица 6

	DSP56824	DSP56F801	DSP56F803	DSP56F805	DSP56F80
Тактовая частота, МГц	70	80	80	80	80
Производительность, MIPS	35	40	40	40	40
Интерфейсы	2SPI, SCI, SSI	SCI, SPI	2SCI, SPI, CAN	2SCI, SPI, CAN	2SCI, SPI, CAN
АЦП	-	2 четырех-кан. 12-разрядн.	2 четырех-кан. 12-разрядн.	2 четырех-кан. 12-разрядн.	4 четырех-кан. 12-разрядн.
ШИМ-генераторы	-	6-канальн. 15-разр.	6-канальн. 15-разр.	2 6-канальн. 15-разр.	2 6-канальн. 15-разр.
Прочие функциональные особенности	ФАПЧ, 3-кан. таймер	НВИ*, ФАПЧ, 2-кан.таймер	НВИ*, ФАПЧ, 5-кан.таймер	НВИ*, ФАПЧ, 6-кан. таймер	НВИ*, ФАПЧ, 6-кан.таймер

. Этот набор встроенных устройств существенно снижает цену системы по сравнению с реализацией устройств управления на основе традиционных DSP. Более того, так как встроенные устройства имеют заранее определенный интерфейс с ядром DSP (в отличие от внешних периферийных устройств), то упрощаются разработка системы, программирование и управление периферийными устройствами. Таким образом, время разработки программ сокращается. Микросхема DSP56F805 содержит следующие периферийные блоки:

- два шестиканальных ШИМ-генератора (PWMA & PWMB) с привязкой импульсов к центру или краю временного интервала, программированием длительности "мертвого времени" и защитой в случае возникновения аварийных режимов работы; каждый генератор снабжен тремя сенсорами тока и четырьмя входами аварийного отключения,

- два 12-разрядных АЦП с одновременной выборкой, снабженные входными четырехканальными мультиплексорами,

- два квадратурных (синусно-косинусных) декодера (Quad Dec0 & Quad Dec1), каждый с четырьмя входами (или два дополнительных четырехканальных таймера A&B),
- два четырехканальных таймера общего применения с шестью входами: таймер С с двумя входами и таймер D с четырьмя входами,
- контроллер CAN интерфейса A/B с двухвыводными портами приемопередатчиков,
- два двухпроводных последовательных коммуникационных интерфейса (SCI0 & SCI1) или 4 дополнительных линии GPIO,
- последовательный интерфейс периферии (SPI) с настраиваемым четырехпроводным портом или четыре дополнительных линии GPIO,
- сторожевой таймер контроля функционирования процессора,
- два программируемых входа внешних прерываний,
- четырнадцать программируемых и восемнадцать мультиплексированных универсальных портов ввода/вывода (GPIO),
- вход принудительного сброса процессора,
- порт JTAG/OnCE™ (встроенного эмулятора) для отладки, не зависящей от тактовой частоты процессора,
- программируемый генератор с ФАПЧ для формирования тактовой частоты ядра DSP.

В системах управления, как правило, интенсивно используются прерывания от внешних устройств и внутренних периферийных модулей. Обычно микроконтроллеры поддерживают несколько типов внутренних и внешних прерываний и обеспечивают много вариантов маскирования и установки приоритетов.

Обычные DSP обрабатывают только небольшой набор прерываний, которые напрямую взаимодействуют с его ядром. В отличие от них, кристалл DSP56F80x поддерживает большое число прерываний. Хотя число адресуемых прерываний ядра DSP56F8xx мало в сравнении с общим числом источников прерываний, многоуровневая встроенная схема мультиплексирования обеспечивает полную и гибкую поддержку 64 источников прерываний, каждый

из которых может маскироваться и имеет программно устанавливаемый приоритет.

Разработка систем на базе микросхем семейства DSP56800 отличается простотой. Внешняя шина обеспечивает выполнение и отладку прикладных программ, размещенных во внешней памяти. Допускается хранение программ и данных во внешней памяти. Чтобы обеспечить функционирование внешней памяти с различным быстродействием программируемые временные задержки для памяти программ и памяти данных могут устанавливаться раздельно.

Набор команд общего назначения, который используется в микропроцессорах с развитыми режимами адресации и командами обработки битов, дает разработчику возможность просто освоить программирование. Сложности, характерные для DSP с предшествующими архитектурами, не доставят ему беспокойства. Программный стек обеспечивает неограниченное число прерываний и вложений подпрограмм, а также поддержку передачи параметров и локальных переменных. Опытный программист найдет широкий набор команд арифметических операций и различные одинарные и двойные обращения к памяти, выполняющиеся совместно с арифметическими операциями. Эффективная работа трансляторов для микросхем с архитектурой DSP56800 обеспечивается использованием в микросхемах команд общего назначения.

Порт отладки JTAG позволяет отлаживать микросхему в составе законченной системы пользователя. Через порт можно задать точки останова программы, проверить и изменить содержимое регистров и ячеек памяти, выполнить другие действия по отладке системы.

Motorola предлагает полный набор программных и аппаратных средств быстрой разработки и отладки систем, реализованных на кристаллах семейства DSP568xx. Средства разработки включают:

- оценочные платы для каждой модификации микросхемы,
- интегрированную среду отладки "Metrowerks Code Warrior" со встроенным кросс-компилятором языка C.

Программная среда разработки предоставляет программисту гибкое модульное окружение, обеспечивая полное использова-

ние возможностей микросхем. Среда допускает различные конфигурации памяти данных и позволяет создавать перемещаемый код, выполнять символьную отладку, гибко компоновать объектные файлы. Реализованы средства создания архива библиотек прикладных программ.

Motorola разработала новый комплект для разработки встроенного программного обеспечения (Embedded Software Development Kit, SDK), дополняющий существующую среду разработки для DSP568xx. Он формирует программную инфраструктуру, обеспечивающую разработку высокоэффективных программ, полностью переносимых и допускающих повторное использование не только в процессорах семейства DSP568xx, но в будущем и в процессорах с другой архитектурой, поддерживаемой SDK. Этот программный продукт, выпускаемый для цифровых сигнальных процессоров фирмы Motorola, предназначен для ускорения разработки и более быстрого выхода изделий на рынок.

Стандартные микроконтроллеры успешно применяются в устройствах управления общего назначения. Однако невысокая производительность не позволяет использовать их в устройствах с повышенными параметрами. Эта ниша заполняется микросхемами семейства DSP568xx, имеющими производительность DSP и снабженными набором периферийных устройств, которые традиционно используют разработчики систем управления.

Архитектура ядра DSP568xx обеспечивает эффективную цифровую обработку данных и решение задач управления. Такие характеристики этой архитектуры, как высокая производительность и набор команд общего назначения, обеспечивают ей лидирующие позиции в тех областях цифровой обработки сигналов, в которых требуется низкая стоимость и малое энергопотребление. Компактность программ и высокая эффективность компилятора позволяет также снизить стоимость системы за счет уменьшения требуемого объема встроенной памяти.

Микросхемы семейства DSP568xx предназначены для применения в недорогих устройствах. Эти микросхемы ориентированы на применение в бытовой технике, для которой необходима низкая стоимость и не требуются высокие параметры. К таким изделиям относятся:

- специализированные и многоцелевые контроллеры,
- проводные и беспроводные модемы,
- системы беспроводной передачи цифровых сообщений,
- цифровые телефонные автоответчики,
- устройства управления серводвигателями и электродвигателями переменного тока,
- цифровые камеры.

Микросхемы этого семейства имеют производительность специализированных DSP. Благодаря наличию набора встроенных периферийных устройств эти микросхемы отвечают требованиям систем управления. Встроенные блоки памяти и периферийные устройства могут существенно снизить стоимость системы, потому что в этом случае уменьшается число внешних компонентов.

8. Архитектуры микропроцессорных систем

8.1. Системы с централизованным, децентрализованным и комбинированным управлением

В МПС с централизованным управлением задача обработки входных сигналов X_1, \dots, X_n с целью формирования управляющих воздействий Y_1, \dots, Y_m решается микропроцессорным устройством (МПУ), включающим МП и элементы памяти, которые соединены каналами связи через цифро-аналоговые преобразователи (ЦАП), исполнительные устройства (ИУ) с объектом (или объектами) управления (ОУ). Обратная связь о состоянии ОУ обеспечивается сигналами от ОУ, поступающими на МПУ через аналого-цифровые преобразователи (АЦП).

Общая структурная схема для этого случая показана на рис. 28. Если осуществляется управление одним, но сложным многомерным объектом (роботом, прокатным станом, доменной печью, самолетом, космическим летательным аппаратом и т. п.), то такая система является связанной. Если же решается задача управления совокупностью независимых по управляемым параметрам одномерных объектов, то система является несвязанной.

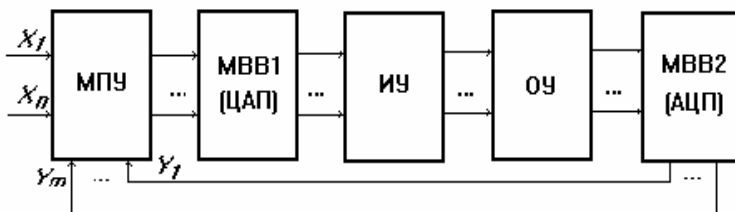


Рис. 28.

В системах с децентрализованным управлением в каждый контур управления включается автономное МПУ. Структурная схема системы с децентрализованным управлением приведена на рис. 29, где МПУ размещены в непосредственной близости от объекта управления ОУ или встроены в него и функционально ориентированы на решение конкретных задач. В качестве МПУ широко применяется программируемые регулирующие микроконтроллеры.

В децентрализованных системах МПУ могут вводиться для передачи ему функций диспетчера либо отсутствовать совсем. В этом случае реализуется комбинированное управление. В комбинированных системах используется обе перечисленные структуры управления.

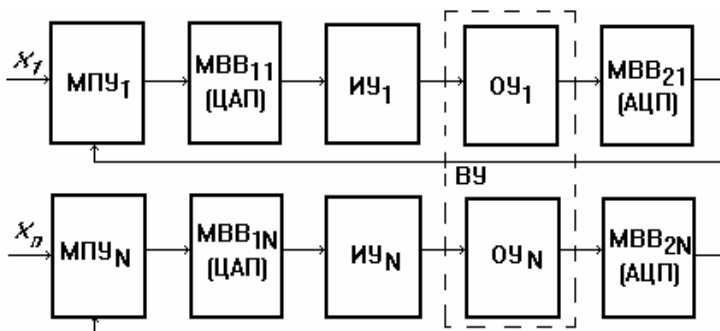


Рис. 29.

Выбор структуры управления в МПС, построенных на базе МК БИС, зависит от многих взаимосвязанных факторов, важнейшими из которых являются стоимость и надежность систем,

их живучесть, гибкость, способность работать в масштабе реального времени.

Специфика конкретных задач управления показывает, что применение принципа децентрализованного (распределенного) управления при построении МПС в техническом и экономическом плане имеют преимущества по сравнению с другими структурами МПС.

8.2. Системы с перестраиваемой структурой

Задачи, решаемые МПС, могут зависеть от характера входных воздействий, поступающих в систему. Так, например, управление роботом может осуществляться по разным алгоритмам в зависимости от результата решения задачи распознавания представленного роботу объекта. В этом и подобных случаях структура МПС оказывается переменной. В микропроцессорных системах она перестраивается программно.

Пример структурной схемы перестраиваемой МПС управления роботом в общем виде показана на рис. 30.

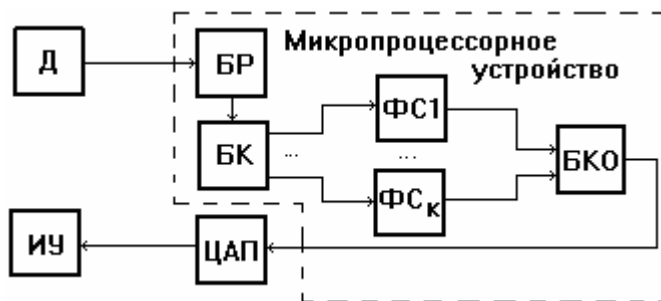


Рис. 30

В состав системы входят: цифровой датчик визуальной информации Д, информация с которого поступает в блок распознавания изображения БР, формирующий электрический логотип изображения. Блок коммутации алгоритмов управления БК производит выбор и обработку программы обслуживания сформированного логотипа. С помощью сигналов формирователей управляю-

щих сигналов $\Phi C_1, \dots, \Phi C_k$ блок коммутации выхода БКВ выдает на выход МПУ исполнительный сигнал, который через ЦАП подается на исполнительное устройство ИУ.

Осуществимость перестройки МПС, выполняемой в реальном масштабе времени на программном уровне, является следствием применения в автоматических системах высокопроизводительных МПС, на которые возлагаются задачи обработки больших потоков информации, связанной со статистическим экспресс-анализом случайных сигналов, их идентификацией, классификацией, распознаванием изображений и т. п. Это в конечном счете существенно улучшает показатели качества управления системой.

8.3. Системы с резервированием

Одним из путей увеличения отказоустойчивости МПС является резервирование. Резервирование подразделяется на аппаратное, программное и информационное.

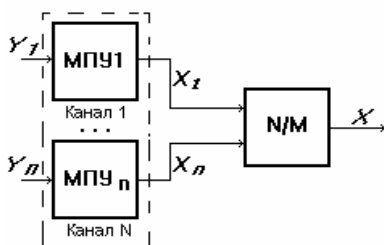


Рис.31

Табл. 7

Y_1	Y_2	Y_3	X
-	-	-	-
-	-	+	-
-	+	-	-
-	+	+	+
+	-	-	-
+	-	+	+
+	+	-	+
+	+	+	+

Распространенными методами аппаратного резервирования являются методы, основанные на мажоритарной обработке и обработке с переключением каналов. При мажоритарной обработке МПС состоит из n независимых каналов обработки информации (рис. 31) и остается в рабочем состоянии до тех пор, пока сохраняют работоспособность t из n каналов. Например, в системе «2 из 3» работоспособное состояние канала определяется из табл. 7, в

которой значками «+» и «-» определено соответственно работоспособное и неработоспособное состояния канала.

В МПС с переключением каналов избыточные (резервные) каналы обработки информации включаются в работу только после выхода из строя основного или ранее замененного канала. В таких системах имеются дополнительные блоки опознавания (БО) неисправных каналов и их переключения БП (рис.32).

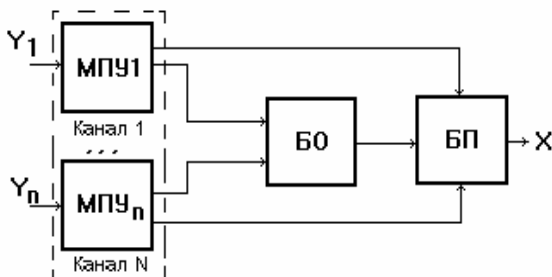


Рис.32

8.4. Иерархические системы

Сложные объекты управления (самолеты, космические аппараты, прокатные станы, роторные конвейерные линии и т. п.) представляют собой совокупность взаимосвязанных многорежимных управляемых систем, объединенных единой системой управления. Основополагающими принципами, определяющими структуру МПС управления подобными объектами, является иерархичность, независимость управления по уровням иерархии и информационная замкнутость. Обобщенная структура иерархической МПС показана на рис.33.

Особенность микроконтроллера проявляется в том, что на его выходе не используется мультиплексирование (число ЦАП равно числу выходных цепей контроллера). Такое построение контроллера связано с необходимостью запоминания каждого значения управляющего сигнала после останова вычислительного процесса. Учитывая, что в распределенной МПС число выходных сигналов невелико, затраты на ЦАП оказываются относительно небольшими.

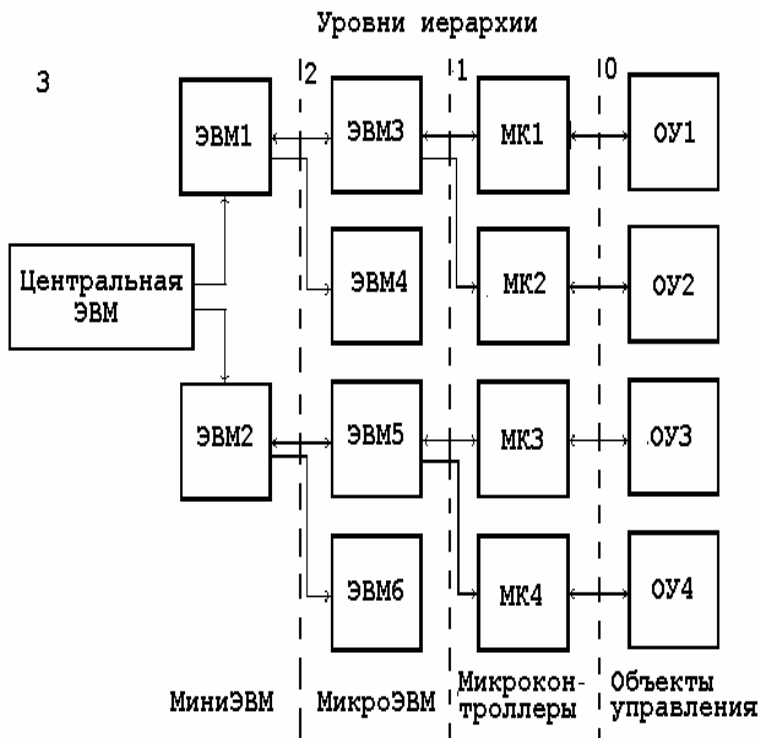


Рис.33

Преобразователи аналоговых сигналов в код позволяют сопрягать микроконтроллер с непрерывными и дискретными датчиками, с исполнительными механизмами пропорционального, позиционного, интегрирующего и другого действия, а также с различными устройствами дискретного и логического управления .

В микроконтроллере может быть применен микропроцессор как с аппаратным, так и с микропрограммным принципом управления. Аппаратное управление основано на внутреннем микропрограммировании. При аппаратном управлении система команд микропроцессора является фиксированной. Она реализована во внутренних жестких электрических связях в кристалле МП и не может быть изменена разработчиком системы. Микропрограммное

управление основано на внешнем микропрограммировании (набор команд может быть нефиксированным и изменяться разработчиком системы).

Особенностью программного обеспечения микроконтроллера является то, что большая часть его памяти программируется на заводе - изготовителе. В нем отсутствуют обычные средства ввода и отладки программ, а также модули сопряжения с ними. Указанные особенности позволяют упростить микроконтроллер и сделать рентабельным его применение для обработки сравнительно небольших массивов информации.

Пульт оператора в микроконтроллере используется для установки требуемой конфигурации регулирующего контура, выбора алгоритма управления, контроля значений технологических переменных, оперативного вмешательства в процесс управления и других целей. Программное обеспечение состоит из программ: диспетчера (координирующего весь вычислительный процесс), рабочих, обслуживания пульта и диагностических. Для программирования используется, как правило, десятичный код, набираемый на панели пульта.

Все алгоритмы микроконтроллера достаточно универсальны и в функциональном отношении эквивалентны типовым звеньям МПС управления объектом или типовой «связке» таких звеньев. Возможности микроконтроллера характеризуют, используя понятие виртуальной (кажущейся) структуры. Виртуальная структура описывает свойства контроллера в традиционных для МПС управления понятиях, основными из которых являются каналы управления, с системной точки зрения эквивалентные отдельному прибору или типовому сочетанию приборов непрерывной системы управления, и конфигурация, определяющая систему связи каналов со входами и выходами контроллера, а также варианты взаимодействия каналов.

По оценкам специалистов, существует ограниченное число (ориентировочно 20 - 25) алгоритмов, комбинация которых позволяет автоматизировать управление процессами и объектами практически любой степени сложности. Эти алгоритмы, оформленные в виде библиотеки программ, хранятся в постоянной памяти и могут быть использованы в любом заданном сочетании. Среди про-

граммируемых регулирующих микроконтроллеров особое место занимают однокристалльные микроконтроллеры, выпускаемые серийно. По степени универсальности использования их подразделяют на специализированные, работающие по жесткой программе, и широкого применения, программа действия которых заносится во внешнее запоминающее устройство и может изменяться самим пользователем или по картам-заказам, составленным пользователем .

Примерами однокристалльных перепрограммируемых микроконтроллеров являются контроллеры серии K145. Это цифровые структуры последовательного действия, использующие принцип многоуровневого программирования. Однокристалльные микроконтроллеры адаптируются к внешним устройствам как по формату управляющих команд, так и по временным характеристикам. Для реализации множества задач управления в таких контроллерах используется специальная система команд, обеспечивающая управление внешними устройствами и выполнение программы. Список команд позволяет организовать как разомкнутую систему управления объектами по жесткой программе, так и замкнутую с большой сетью внутри программных ветвлений в соответствии с условиями, задаваемыми по времени и состоянию датчиков.

На втором уровне иерархии находятся серийные микроЭВМ, которые обеспечивают управление группой функционально связанных объектов. На этом уровне, соответствующем локальному управлению, применяют серийные микроЭВМ многофункционального назначения.

Третий уровень включает управляющие устройства, реализованные на базе мини-ЭВМ, которые координируют работу группы локальных систем.

На четвертом уровне располагается центральная управляющая ЭВМ, которая является высшим координирующим органом в данной структуре.

8.5. Однопроцессорная МПС типа «Общая шина»

Однопроцессорная МПС содержит в своем составе один микропроцессор, один или несколько модулей памяти, систему

связи с объектами управления или измерительными объектами, состоящую из устройств ввода-вывода, которые в общем виде носят название контроллеров ввода - вывода (КВВ), внешних устройств, в качестве которых используются датчики и исполнительные механизмы, и систему ввода - вывода, в которую входят УВВ и ВУ. Перечисленные компоненты МПС, типовая структурная схема которой представлена на рис.34, связаны с системной магистралью посредством соответствующих интерфейсов. В свою очередь магистраль состоит из шины адресов, шины данных и управляющей шины.

Такая система на практике получила название «общая шина».

В большинстве случаев МПС используют магистрально - модульный принцип построения. Суть этого принципа заключается в том, что отдельные блоки являются функционально законченными модулями со своими встроенными схемами управления, выполненными в виде одного или нескольких кристаллов БИС или СБИС.

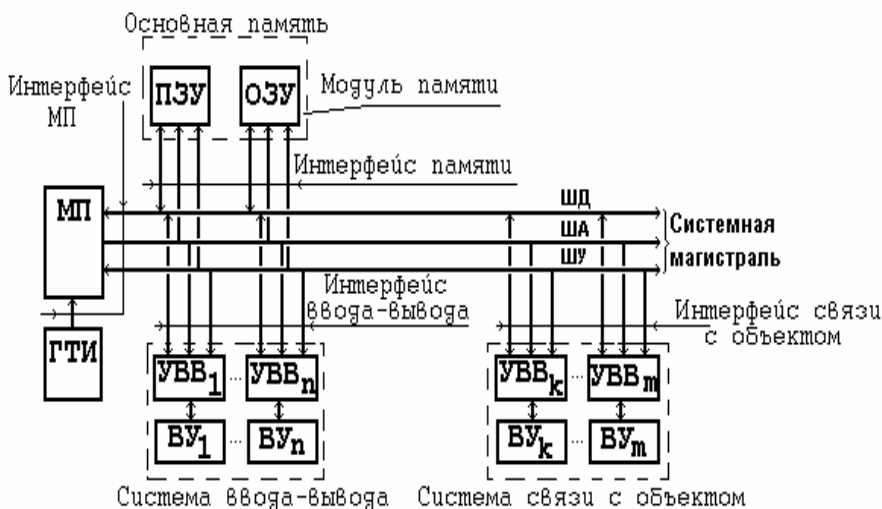


Рис. 34

Межмодульные связи и обмен информацией между модулями осуществляются посредством шин (магистралей), к которым

имеют доступ все основные модули системы. В МПС в каждый данный момент времени возможен обмен информацией только между двумя модулями системы.

Ситуация, при которой три или более модуля требуют одновременного доступа к одной магистрали, является недопустимой, так как она приводит к появлению конфликта. Поэтому обмен информацией в магистрально-модульных системах производится, как правило, путем разделения (арбитража) во времени управления модулями системы магистралей.

Особенностью магистрально-модульного принципа построения МПС является необходимость информационно - логической совместимости модулей. При выполнении этого принципа достигается оптимальная межмодульная передача информации. Он реализуется путем использования единых способов представления информации, алгоритмов управления обменом, форматов команд управления обменом и способов синхронизации, то есть выполнения определенных электрических и конструктивных требований при построении интерфейсов.

При построении МПС наибольшее применение получила трехшинная структура, содержащая шину адреса ША, двунаправленную шину данных ШД и шину управления ШУ. Как видно из рис.34, МПС предполагает наличие общего сопряжения (общего или единого интерфейса) для модулей памяти — постоянных и оперативных запоминающих устройств (ПЗУ и ОЗУ) и периферийных устройств — внешних запоминающих устройств (ВЗУ) и УВВ.

Модуль памяти ОП (или основная память) МПС содержит постоянное запоминающее устройство ПЗУ и оперативное запоминающее устройство ОЗУ. В большинстве системах ОП физически реализуется в виде многоуровневой иерархической системы. Верхние уровни памяти строятся на основе полупроводниковых ПЗУ и ОЗУ, а нижние - на основе магнитных ВЗУ.

Основная память является одним из важнейших компонентов любой МПС и предназначена для хранения программ и данных, используемых или генерируемых выполняющимися программами. Главная характеристика основной памяти - объем или емкость. В

МПС используются две единицы измерения емкости основной памяти - байт и слово.

Байт - это последовательность из 8 бит, рассматриваемая как один элемент данных или памяти. Биты в байте нумеруются от 0 до 7 в порядке справа налево, причем бит 0 считает младшим, а бит 7 - старшим в байте (рис.35, а).

Каждый байт памяти имеет свой адрес. Так, например, в МП с 16 - разрядной ША адреса байтов могут иметь значения в диапазоне целых чисел от 0 до $2^{16} - 1$ (до 65535). Следовательно, общее число всех адресов - адресное пространство ОП, составляет $2^{16} = 65\ 536 = 64\text{К}$. Эта величина, указываемая в технических характеристиках, называется размером адресного пространства, выраженным числом адресов байт. Иногда ее называют также емкостью ОП.



Рис. 35

Другая, более крупная единица измерения емкости ОП - слово, которое состоит из двух байт и рассматривается как один элемент. Из двух байт, составляющих слово (рис.35, б), младший имеет четный адрес, а старший (на единицу больший) - нечетный адрес. Адресом слова считается адрес младшего байта, входящего в слово. Следовательно, слово всегда имеет четный адрес.

Принцип четной адресации слов в МПС должен строго соблюдаться программистом, так как при его нарушении многие команды будут выполняться с ошибкой.

Чтобы дать более полное представление о МПС, помимо высокой производительности необходимо назвать и другие отличительные особенности. Прежде всего это необычные архитектуру-

ные решения, направленные на повышение производительности (работа с векторными операциями, организация быстрого обмена сообщениями между процессорами или организация глобальной памяти в многопроцессорных системах и др.).

Понятие архитектуры высокопроизводительной системы является достаточно широким, поскольку под архитектурой можно понимать и способ параллельной обработки данных, используемый в системе, и организацию памяти, и топологию связи между процессорами, и способ исполнения системой арифметических операций. Эти вопросы освещаются ниже.

8.6. Архитектуры с параллельной обработкой данных

В 1966 году М.Флинном был предложен подход к классификации архитектур вычислительных систем. В основу было положено понятие потока, под которым понимается последовательность элементов, команд или данных, обрабатываемая процессором. Соответствующая система классификации основана на рассмотрении числа потоков инструкций и потоков данных. Она содержит четыре архитектурных класса:

SISD = Single Instruction Single Data,

MISD = Multiple Instruction Single Data ,

SIMD = Single Instruction Multiple Data,

MIMD = Multiple Instruction Multiple Data.

SISD - одиночный поток команд и одиночный поток данных. К этому классу относятся последовательные МПС, которые имеют один центральный процессор, способный обрабатывать только один поток последовательно исполняемых инструкций (рис. 36). В настоящее время практически все высокопроизводительные системы имеют более одного центрального процессора, однако, каждый из них выполняют несвязанные потоки инструкций, что делает такие системы комплексами SIMD-систем, действующих на разных пространствах данных.



Рис. 36

Для увеличения скорости обработки команд и скорости выполнения арифметических операций может применяться конвейерная обработка. В случае векторных систем векторный поток данных следует рассматривать как поток из одиночных неделимых векторов. Примерами компьютеров с архитектурой SISD являются большинство рабочих станций Compaq, Hewlett-Packard и Sun Microsystems.

MISD - множественный поток команд и одиночный поток данных. Теоретически в этом типе МПС множество инструкций должно выполняться над единственным потоком данных (рис.37).



Рис. 37

Определение подразумевает наличие в архитектуре многих процессоров, обрабатывающих один и тот же поток данных. Однако развитие эта архитектура МПС до сих пор не получила.

По ряду признаков к этому классу можно отнести конвейерные МПС, однако это не нашло окончательного признания на практике.

SIMD - одиночный поток команд и множественный поток данных (рис.38). Эти системы обычно имеют большое количество процессоров (от 1024 до 16384), которые могут выполнять одну и ту же инструкцию относительно разных данных в жесткой конфигурации. Единственная инструкция параллельно выполняется над многими элементами данных.

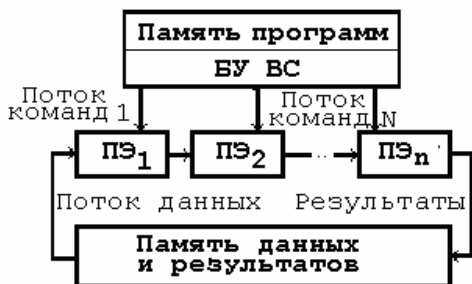


Рис. 38

Примерами SIMD машин являются системы CPP DAP, Gamma II и Quadrics Apemille. Другим подклассом SIMD-систем являются векторные МПС. Они манипулируют массивами сходных данных подобно тому, как скалярные системы обрабатывают отдельные элементы таких массивов. Это делается за счет использования специально сконструированных векторных центральных процессоров. Когда данные обрабатываются посредством векторных модулей, результаты могут быть выданы на один, два или три такта частоты генератора (такт частоты генератора является основным временным параметром системы).

При работе в векторном режиме векторные процессоры обрабатывают данные практически параллельно, что делает их в несколько раз более быстрыми, чем при работе в скалярном режиме. Примерами систем подобного типа является, например, Hitachi S3600.

MIMD - множественный поток команд и множественный поток данных (рис. 39). Эти системы параллельно выполняют несколько потоков инструкций над различными потоками данных. В отличие от многопроцессорных SISD-систем, упомянутых выше,

команды и данные связаны, потому что они представляют различные части одной и той же выполняемой задачи. Например, MIMD-системы могут параллельно выполнять множество подзадач, с целью сокращения времени выполнения основной задачи.



Рис. 39

Наличие большого разнообразия попадающих в данный класс систем, делает классификацию Флинна не полностью адекватной. Действительно и четырехпроцессорный SX-5 компании NEC и тысячепроцессорный Cray T3E оба попадают в этот класс. Это заставляет использовать другой подход к классификации, иначе описывающий классы МПС. Основная идея такого подхода может состоять, например, в следующем. Считаем, что множественный поток команд может быть обработан двумя способами: либо одним конвейерным устройством обработки, работающем в режиме разделения времени для отдельных потоков, либо каждый поток обрабатывается своим собственным устройством. Первая возможность используется в MIMD-системах, которые обычно называют конвейерными или векторными, вторая – в параллельных системах.

В основе векторных МПС лежит концепция конвейеризации, т.е. явного сегментирования арифметического устройства на отдельные части, каждая из которых выполняет свою подзадачу для пары операндов. В основе параллельной системы лежит идея использования для решения одной задачи нескольких процессо-

ров, работающих сообща, причем процессоры могут быть как скалярными, так и векторными.

8.6.1. SMP архитектура

SMP архитектура (symmetric multiprocessing) - симметричная многопроцессорная архитектура (рис.40). Главной особенностью систем с архитектурой SMP является наличие общей физической памяти, разделяемой всеми процессорами.

Память является способом передачи сообщений между процессорами, при этом все вычислительные устройства при обращении к ней имеют равные права и одну и ту же адресацию для всех ячеек памяти. Поэтому SMP архитектура называется симметричной. Последнее обстоятельство позволяет очень эффективно обмениваться данными с другими вычислительными устройствами.

SMP-система строится на основе высокоскоростной системной шины (SGI PowerPath, Sun Gigaplane, DEC TurboLaser), к слотам которой подключаются функциональные блоки трех типов: процессоры (ЦП), операционная система (ОП) и подсистема ввода/вывода (I/O). Для подсоединения к модулям I/O используются уже более медленные шины (PCI, VME64).

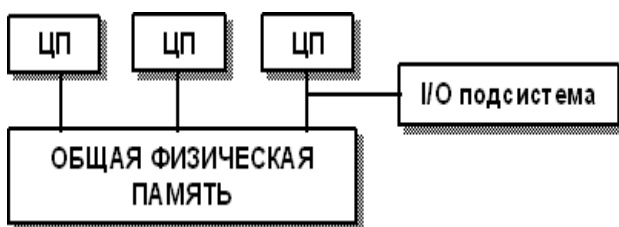


Рис. 40

Наиболее известными SMP-системами являются SMP-сервера и рабочие станции на базе процессоров Intel (IBM, HP, Compaq, Dell, ALR, Unisys, DG, Fujitsu и др.). Вся система работает под управлением единой ОС (обычно UNIX-подобной, но для Intel-платформ поддерживается Windows NT). ОС автоматически

(в процессе работы) распределяет процессы по процессорам, но иногда возможна и явная привязка.

Основные преимущества SMP-систем:

- простота и универсальность для программирования. Архитектура SMP не накладывает ограничений на модель программирования, используемую при создании приложения: обычно используется модель параллельных ветвей, когда все процессоры работают абсолютно независимо друг от друга - однако, можно реализовать и модели, использующие межпроцессорный обмен. Использование общей памяти увеличивает скорость такого обмена, пользователь также имеет доступ сразу ко всему объему памяти. Для SMP-систем существуют сравнительно эффективные средства автоматического распараллеливания.

- легкость в эксплуатации. Как правило, SMP-системы используют систему охлаждения, основанную на воздушном кондиционировании, что облегчает их техническое обслуживание.

- относительно невысокая цена.

К недостаткам архитектуры можно отнести:

- системы с общей памятью, построенные на системной шине, плохо масштабируемы

Этот важный недостаток SMP-системы не позволяет считать их по-настоящему перспективными. Причины плохой масштабируемости состоят в том, что в данный момент шина способна обрабатывать только одну транзакцию, вследствие чего возникают проблемы разрешения конфликтов при одновременном обращении нескольких процессоров к одним и тем же областям общей физической памяти. Вычислительные элементы начинают друг другу мешать. Когда произойдет такой конфликт, зависит от скорости связи и от количества вычислительных элементов. В настоящее время конфликты могут происходить при наличии 8-24-х процессоров. Кроме того, системная шина имеет ограниченную (хоть и высокую) пропускную способность и ограниченное число слотов. Все это с очевидностью препятствует увеличению производительности при увеличении числа процессоров и числа подключаемых пользователей.

В реальных системах можно использовать не более 32 процессоров. Для построения масштабируемых систем на базе SMP используются кластерные или NUMA-архитектуры.

8.6.2. MPP архитектура

MPP архитектура (massive parallel processing) - массивно-параллельная архитектура. Главная особенность такой архитектуры состоит в том, что память физически разделена. В этом случае система строится из отдельных модулей, содержащих процессор, локальный банк операционной памяти (ОП), два коммуникационных процессора (рутера) или сетевой адаптер, иногда - жесткие диски и/или другие устройства ввода/вывода. Один рутер используется для передачи команд, другой - для передачи данных. По сути, такие модули представляют собой полнофункциональные системы (рис.41).

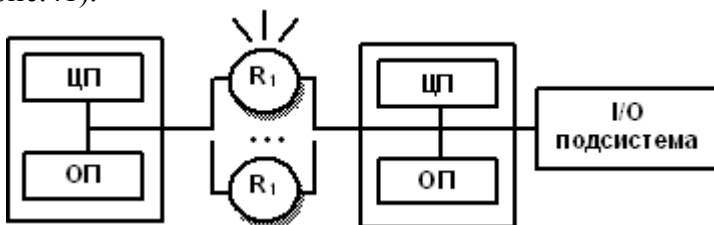


Рис. 41

Доступ к банку ОП из данного модуля имеют только процессоры (ЦП) из этого же модуля. Модули соединяются специальными коммуникационными каналами. Пользователь может определить логический номер процессора, к которому он подключен, и организовать обмен сообщениями с другими процессорами.

Используются два варианта работы операционной системы (ОС) на машинах MPP архитектуры. В одном полноценная операционная система (ОС) работает только на управляющей машине (front-end), на каждом отдельном модуле работает сильно урезанный вариант ОС, обеспечивающий работу только расположенной в нем ветви параллельного приложения. Во втором варианте на каждом модуле работает полноценная UNIX-подобная ОС, устанавливаемая отдельно на каждом модуле.

Главным преимуществом систем с раздельной памятью является хорошая масштабируемость: в отличие от SMP-систем в машинах с раздельной памятью каждый процессор имеет доступ только к своей локальной памяти, в связи с чем не возникает необходимости в потактовой синхронизации процессоров. Практически все рекорды по производительности на сегодняшний день устанавливаются на системах именно такой архитектуры, состоящих из нескольких тысяч процессоров (ASCI Red, ASCI Blue Pacific).

К недостаткам можно отнести:

- отсутствие общей памяти заметно снижает скорость межпроцессорного обмена, поскольку нет общей среды для хранения данных, предназначенных для обмена между процессорами. Требуется специальная техника программирования для реализации обмена сообщениями между процессорами;

- каждый процессор может использовать только ограниченный объем локального банка памяти;

- вследствие указанных архитектурных недостатков требуются значительные усилия для того, чтобы максимально использовать системные ресурсы. Именно этим определяется высокая цена программного обеспечения для массивно-параллельных систем с раздельной памятью.

Системами с раздельной памятью являются суперкомпьютеры MBC-1000, IBM RS/6000 SP, SGI/CRAY T3E, системы ASCI, Hitachi SR8000, системы Parsytec.

Системы последней серии CRAY T3E от SGI, основанные на базе процессоров Dec Alpha 21164 с пиковой производительностью 1200 Мфлопс/с (CRAY T3E-1200), способны масштабироваться до 2048 процессоров.

8.6.3. Гибридная архитектура

Гибридная архитектура (NUMA - nonuniform memory access) характеризуется неоднородным доступом к памяти. Она воплощает в себе удобства систем с общей памятью и относительную дешевизну систем с раздельной памятью. Суть этой архитектуры поясним на примере архитектуры сервера NUMA Q-2000, разработанного фирмами IBM и Sequent в 1999 году. Он имеет

особую организацию памяти, а именно: память является физически распределенной по различным частям системы, но логически разделяемой, так что пользователь видит единое адресное пространство. Система состоит из однородных базовых модулей (плат), состоящих из

небольшого числа процессоров и блока памяти. Модули объединены с помощью высокоскоростного коммутатора. Поддерживается единое адресное пространство, аппаратно поддерживается доступ к удаленной памяти, т.е. к памяти других модулей. При этом доступ к локальной памяти осуществляется в несколько раз быстрее, чем к удаленной. По существу архитектура NUMA является MPP (массивно-параллельная архитектура) архитектурой, где в качестве отдельных вычислительных элементов берутся SMP (симметричная многопроцессорная архитектура) узлы.

Структурная схема системы с гибридной сетью представлена на рис. 42. Она содержит четыре процессора, которые связываются между собой при помощи кроссбара в рамках одного SMP узла. Узлы связаны сетью типа "бабочка" (Butterfly):

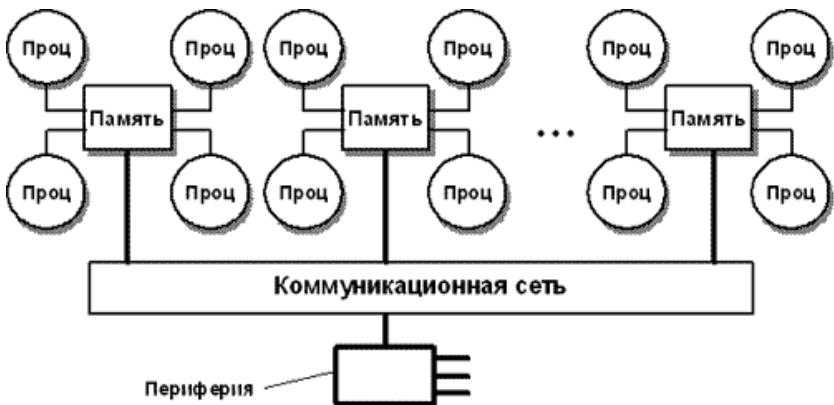


Рис. 42

Впервые идею гибридной архитектуры предложил Стив Воллох и воплотил в системах серии Exemplar. Вариант Воллоха - система, состоящая из 8-ми SMP узлов. Фирма HP купила идею и реализовала на суперкомпьютерах серии SPP. Идею подхватил Сеймур Крей (Seymour R. Cray) и добавил новый элемент - когерентный кэш, создав так называемую архитектуру cc-NUMA (Cache Coherent Non-Uniform Memory Access), которая расшифровывается как "неоднородный доступ к памяти с обеспечением когерентности кэшей". Он ее реализовал на системах типа Origin.

Понятие когерентности кэшей описывает тот факт, что все центральные процессоры получают одинаковые значения одних и тех же переменных в любой момент времени. Действительно, поскольку кэш-память принадлежит отдельной системе, а не всей многопроцессорной системе в целом, данные, попадающие в кэш одной МПС, могут быть недоступны другой. Чтобы избежать этого, следует провести синхронизацию информации, хранящейся в кэш-памяти процессоров.

Для обеспечения подобной когерентности кэшей существуют несколько возможностей:

- использовать механизм отслеживания шинных запросов (snoopy bus protocol), в котором кэши отслеживают переменные, передаваемые к любому из центральных процессоров и, при необходимости, модифицируют собственные копии таких переменных;
- выделять специальную часть памяти, отвечающую за отслеживание достоверности всех используемых копий переменных.

Наиболее известными системами архитектуры cc-NUMA являются: HP 9000 V-class в SCA-конфигурациях, SGI Origin3000, Sun HPC 15000, IBM/Sequent NUMA-Q 2000. На настоящий момент максимальное число процессоров в cc-NUMA-системах может превышать 1000 (серия Origin3000). Обычно вся система работает под управлением единой ОС, как в SMP.



Возможны также варианты динамического "подразделения" системы, когда отдель-

ные "разделы" системы работают под управлением разных ОС.

8.6.4. PVP архитектура

PVP (Parallel Vector Process) - параллельная архитектура с векторными процессорами. Основным признаком PVP-систем является наличие специальных векторно-конвейерных процессоров, в которых предусмотрены команды однотипной обработки векторов независимых данных, эффективно выполняющиеся на конвейерных функциональных устройствах. Как правило, несколько таких процессоров (от 1 до 16) работают одновременно с общей памятью (аналогично SMP) в рамках многопроцессорных конфигураций. Несколько таких узлов могут быть объединены с помощью коммутатора (аналогично MPP). Поскольку передача данных в векторном формате осуществляется намного быстрее, чем в скалярном (максимальная скорость может составлять 64 Гб/с, что на 2 порядка быстрее, чем в скалярных системах), то проблема взаимодействия между потоками данных при распараллеливании становится несущественной. И то, что плохо распараллеливается на скалярных машинах, хорошо распараллеливается на векторных.

Таким образом, системы PVP архитектуры могут являться системами общего назначения (general purpose systems). Однако, поскольку векторные процессоры весьма дороги, эти системы не будут являться общедоступными.

Наиболее популярными системами PVP архитектуры являются:



- CRAY SV-2, SMP архитектура. Пиковая производительность системы в стандартной конфигурации может составлять десятки терафлопс.

- NEC SX-6, NUMA архитектура. Пиковая производительность системы может достигать 8 Тфлопс, производительность 1 процессора составляет 8 Гфлопс. Система масштабируется до 128 узлов.

- Fujitsu-VPP5000 (vector parallel processing)), MPP архитектура. Производительность 1 процессора составляет 9.6 Гфлопс, пиковая производительность системы может достигать 1249 Гфлопс, максимальная емкость памяти - 8 Тб. Система масштабируется до 512 узлов.



8.6.5. Кластерная архитектура

Кластер представляет собой две или больше систем (часто называемых узлами), объединяемых при помощи сетевых технологий на базе шинной архитектуры или коммутатора и предстающих перед пользователями в качестве единого информационно-вычислительного ресурса. В качестве узлов кластера могут быть выбраны серверы, рабочие станции и даже обычные персональные компьютеры.

Преимущество кластеризации для повышения работоспособности становится очевидным в случае сбоя какого-либо узла: при этом другой узел кластера может взять на себя нагрузку неисправного узла, и пользователи не заметят прерывания в доступе. Возможности масштабируемости кластеров позволяют многократно увеличивать производительность приложений для большего числа пользователей. технологий на базе шинной архитектуры или коммутатора. Такие суперкомпьютерные системы являются самыми дешевыми, поскольку собираются на базе стандартных комплектующих элементов ("off the shelf"), процессоров, коммутаторов, дисков и внешних устройств.

Кластеризация может быть осуществлена на разных уровнях компьютерной системы, включая аппаратное обеспечение, операционные системы, программы-утилиты, системы управления и приложения. Чем больше уровней системы объединены кластерной технологией, тем выше надежность, масштабируемость и управляемость кластера.

Кластеры условно делятся на классы:

Класс I. Класс строится целиком из стандартных деталей, которые продают многие продавцы компьютерных компонент (низкие цены, простое обслуживание, аппаратные компоненты доступны из различных источников).

Класс II. Система имеет эксклюзивные или не широко распространенные детали. Этим можно достичь очень хорошей производительности, но при более высокой стоимости.

Как уже указывалось выше, кластеры могут существовать в различных конфигурациях. Наиболее употребляемыми типами кластеров являются:

- системы высокой надежности,
- системы для высокопроизводительных вычислений,
- многопоточные системы.

Заметим, что границы между этими типами кластеров до некоторой степени размыты, и часто существующий кластер может иметь такие свойства или функции, которые выходят за рамки перечисленных типов. Более того, при конфигурировании большого кластера, используемого как система общего назначения, приходится выделять блоки, выполняющие все перечисленные функции.

Кластеры для высокопроизводительных вычислений предназначены для параллельных расчётов. Эти кластеры обычно собраны из большого числа компьютеров. Разработка таких кластеров является сложным процессом, требующим на каждом шаге аккуратных согласований таких вопросов как инсталляция, эксплуатация и одновременное управление большим числом компьютеров, технические требования параллельного и высокопроизводительного доступа к одному и тому же системному файлу (или файлам) и межпроцессорная связь между узлами и координация работы в параллельном режиме. Эти проблемы проще всего решаются при обеспечении единого образа операционной системы для всего кластера. Однако реализовать подобную схему удаётся далеко не всегда и обычно она обычно применяется лишь для не слишком больших систем.

Многопоточные системы используются для обеспечения единого интерфейса к ряду ресурсов, которые могут со временем

произвольно наращиваться (или сокращаться) в размере. Наиболее общий пример этого представляет собой группа Веб-серверов.

В 1994 году Томас Стерлинг (Sterling) и Дон Беккер (Becker) создали 16-и узловой кластер «Beowulf» из процессоров Intel DX4, соединенных сетью 10Мбит/с Ethernet с дублированием каналов. Кластер возник в центре NASA Goddard Space Flight Center для поддержки необходимыми вычислительными ресурсами проекта Earth and Space Sciences. Проектно-конструкторские работы над кластером быстро превратились в то, что известно сейчас под названием проект Beowulf.

Проект стал основой общего подхода к построению параллельных кластерных компьютеров и описывает многопроцессорную архитектуру, которая может с успехом использоваться для параллельных вычислений. Beowulf-кластер, как правило, является системой, состоящей из одного серверного узла (который обычно называется головным узлом), а также одного или нескольких подчинённых узлов (вычислительных узлов), соединённых посредством стандартной компьютерной сети. Система строится с использованием стандартных аппаратных компонент, таких как ПК, запускаемых под Linux, стандартных сетевых адаптеров (например, Ethernet) и коммутаторов. Нет особого программного пакета, называемого «Beowulf». Вместо этого имеется несколько кусков программного обеспечения, которые многие пользователи нашли пригодными для построения кластеров Beowulf. Beowulf использует такие программные продукты как операционную систему Linux, системы передачи сообщений PVM, MPI, системы управления очередями заданий и другие стандартные продукты. Серверный узел контролирует весь кластер и обслуживает файлы, направляемые к клиентским узлам.

При разработке кластерных систем существует ряд проблем, среди которых можно выделить следующие.

Архитектура кластерной системы (способ соединения процессоров друг с другом) в большей степени определяет ее производительность, чем тип используемых в ней процессоров. Критическим параметром, влияющим на величину производительности такой системы, является расстояние между процессорами. Так, со-

единив вместе 10 персональных компьютеров, мы получим систему для проведения высокопроизводительных вычислений, проблема, однако, будет состоять в нахождении наиболее эффективного способа соединения стандартных средств друг с другом, поскольку при увеличении производительности каждого процессора в 10 раз производительность системы в целом в 10 раз не увеличится.

Рассмотрим для примера задачу построения симметричной 16-ти процессорной системы, в которой все процессоры были бы равноправны. Наиболее естественным представляется соединение в виде плоской решетки, где внешние концы используются для подсоединения внешних устройств (рис.43).

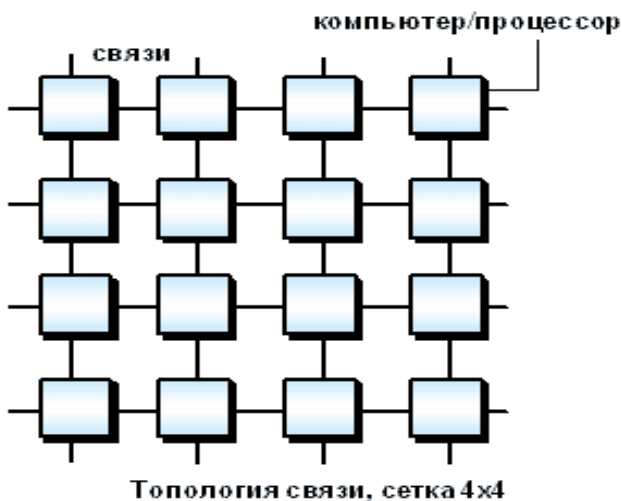


Рис. 43

При таком типе соединения максимальное расстояние между процессорами окажется равным 6 (количество связей между процессорами, отделяющих самый ближний процессор от самого дальнего). Теория же показывает, что если в системе максимальное расстояние между процессорами больше 4, то такая система не может работать эффективно. Поэтому, при соединении 16 процессоров друг с другом плоская схема является не эффективной.

Для получения более компактной конфигурации необходимо решить задачу о нахождении фигуры, имеющей максимальный

объем при минимальной площади поверхности. В трехмерном пространстве таким свойством обладает шар. Но поскольку нам необходимо построить узловую систему, то вместо шара приходится использовать куб (если число процессоров равно 8) или гиперкуб, если число процессоров больше 8. Размерность гиперкуба будет определяться в зависимости от числа процессоров, которые необходимо соединить. Так, для соединения 16 процессоров потребуется 4-х мерный гиперкуб. Для его построения следует взять обычный 3-х мерный куб, сдвинуть в еще одном направлении и, соединив вершины, получить гиперкуб размером 4 (рис. 44). Архитектура гиперкуба является второй по эффективности, но самой наглядной.

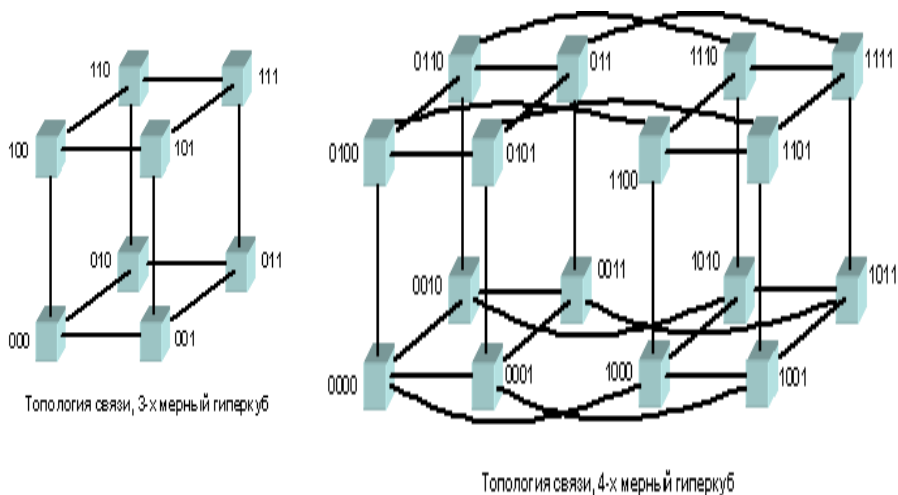


Рис. 44

Используются и другие топологии сетей связи, например,. Наиболее эффективной является архитектура с топологией fat-tree (рис. 45 - архитектура кольца с полной связью по хордам, рис. 46 - кластерная архитектура Fat Free, вид спереди (а) и вид сверху (б)).

. Архитектура fat-tree предложена Лейзерсоном в 1985 году. Процессоры локализованы в листьях дерева, в то время как внутренние узлы дерева скомпонованы во внутреннюю сеть. Поддере-

вья могут общаться между собой, не затрагивая более высоких уровней сети.

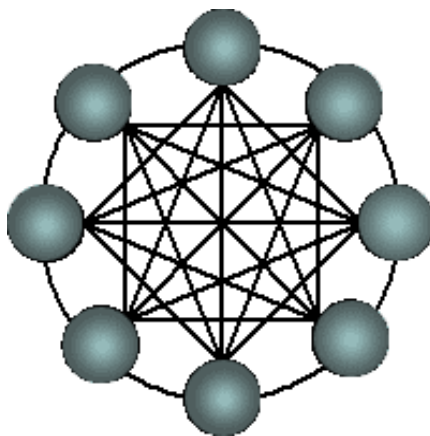


Рис. 45

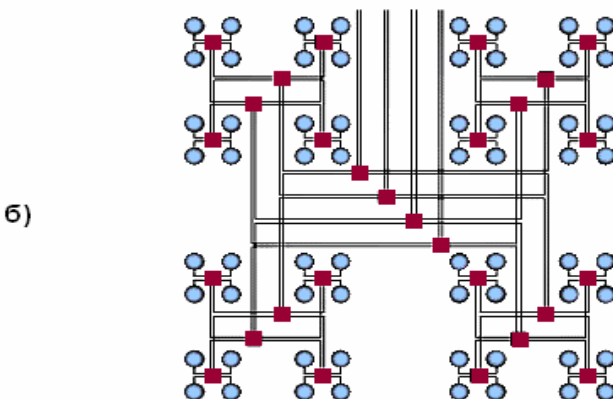
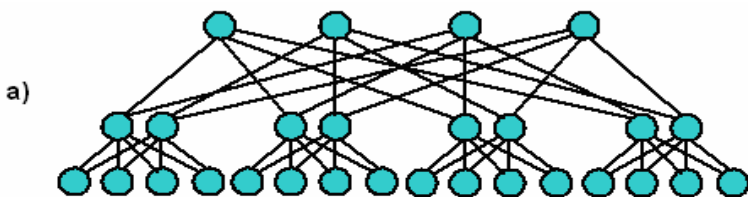


Рис. 46

Поскольку способ соединения процессоров друг с другом больше влияет на производительность кластера, чем тип используемых в ней процессоров, то может оказаться более рентабель-

ным создать систему из большего числа дешевых компьютеров, чем из меньшего числа дорогих.

В кластерах, как правило, используются операционные системы, стандартные для рабочих станций, чаще всего, свободно распространяемые - Linux, FreeBSD, вместе со специальными средствами поддержки параллельного программирования и балансировки нагрузки.

8.6.6. Транспьютеры

Транспьютер – это слово, производное от слов транзистор и компьютер. Транспьютер - это микроэлектронный прибор, объединяющий на одном кристалле микропроцессор, быструю память, интерфейс внешней памяти и каналы ввода-вывода (линки), предназначенные для подключения аналогичных приборов. Прибор спроектирован таким образом, чтобы максимально облегчить построение параллельных вычислительных систем. При соединении транспьютерных элементов между собой требуется минимальное число дополнительных интегральных схем. Связь между транспьютерами осуществляется путем непосредственного соединения линка одного прибора с линком другого. Это позволяет создавать сети с различными топологиями с большим числом элементов.

Производство транспьютеров началось в 1985 году с выпуска 32 – разрядного транспьютера T414 с фоннеймановской КСНК – архитектурой (рис. 47).

Взаимодействие каждого транспьютера с другими транспьютерами и периферийными устройствами осуществляется посредством 4-х коммуникационных каналов связи, имеющих в составе БИС. Для передачи сообщений из внутренней и внекристалльной локальной памяти по последним каналам применяется механизм блочных ПДП-пересылок. Интерфейсы связи и процессор работают одновременно, что приводит лишь к незначительной потере производительности процессора. Использование прямых последовательных коммуникационных каналов делает ненужным арбитраж приоритетов и исключает проблемы, связанные с пропускной способностью шин и их перегрузкой при введении в систему новых процессоров.

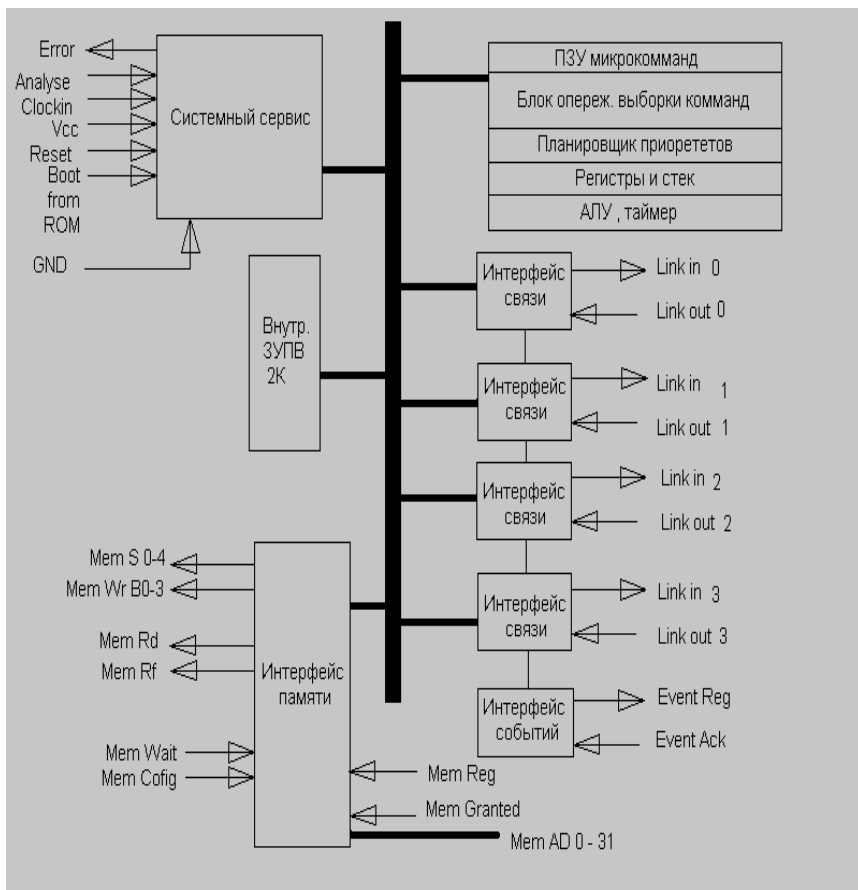


Рис.47.

Каждый исследуемый канал состоит из двух частей, служащих для передачи информации в противоположных направлениях. Пересылка производится в виде последовательностей байтов со скоростью 10-20 Мбит/с, причем каждому байту предшествуют два единичных бита, а завершает передачу один нулевой бит. После передачи байта данных пославший его транспьютер ожидает получения двухбитового подтверждающего сигнала, указывающего на то, что принимающий транспьютер готов к дальнейшему приему информации. Возможен обмен информации между незави-

симотактируемыми системами, если частоты тактирования одинаковы.

Для сопряжения последовательных каналов транспьютера с нетранспьютерными устройствами и интерфейсами связи предусмотрен ряд адаптерных схем. Контроллеры периферийных устройств также могут быть присоединены к имеющейся памяти, посредством которой возможно их обращение ко всему пространству памяти.

Система прерываний в традиционном смысле этого понятия отсутствует, но имеют место аналогичные средства, реализованные в виде двух уровней приоритета, присваиваемых процессам, ожидающим приема по последовательным каналам и запросному входу события.

Транспьютер может быть использован в качестве отдельного самостоятельного устройства, обеспечивающего производительность 10 млн. оп/с. При этом для программирования используется широкий набор стандартных высокоуровневых языков, т.к. архитектура транспьютера ориентирована на эффективное применение компиляции. Для полной реализации возможностей объединения транспьютеров в сети или матрицы при построении высокопроизводительных систем целесообразно применять язык Окками, позволяющий максимальным образом использовать свойства транспьютеров, ориентированные на распараллеливание обработки.

Аппаратные средства транспьютера прямым образом ориентированы на реализацию параллельной обработки и соответствующих информационных пересылок (рис. 48).

Планировщик позволяет одновременно выполнять любое количество параллельных процессов, между которыми распределяется процессорное время. Время переключения процессов составляет менее 1 мкс, а обмен информации между процессами осуществляется посредством блочных передач ввод-вывод памяти.

Активные процессы, ожидающие выполнения, содержатся в связанной очереди рабочих пространств, которая реализована с помощью двух регистров, один указывает на первый процесс в очереди, а второй указывает на последний процесс. Как только выполнение процесса становится возможным, указатель команды за-

поминается в его рабочем пространстве, и из очереди берется для выполнения следующий процесс.

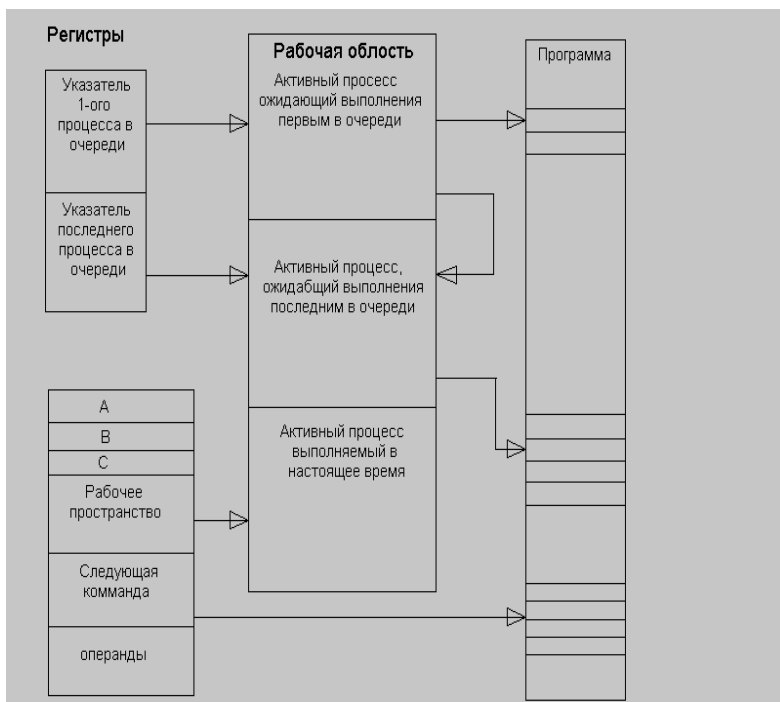


Рис. 48

Транспьютер обеспечивает двухуровневую систему приоритетов. Процессы низкого уровня приоритета должны ожидать выполнения до тех пор, пока в активном состоянии не останется ни одного процесса высокого уровня.

Выполнение процессов низкого уровня разделяется с определенной периодичностью на интервалы времени, чтобы процессорное время равномерно распределялось между задачами, обработка которых требует больших временных затрат. Продолжительность периода квантования составляет 4096 периодов входного текстового сигнала, имеющего частоту 5 МГц, т.е. примерно 800 мкс. Не должна иметь место ситуация, когда весь период квантов был бы занят выполнением процессов высокого уровня.

Подготовка процесса к вводу или выводу заключается в загрузке в вычислительный стек:

- идентификатора канала;
- количества подлежащих пересылке байтов;
- указателя буфера.

Передача сообщения по внутреннему каналу осуществляется путем занесения идентификатора первого процесса, оказавшегося готовым в слово канала и занесения указателя в рабочее пространство. После этого процессор выполняет очередной процесс из планирующей очереди.

Когда оказывается готовым второй процесс, использующий этот же канал, происходит перепись сообщения, ожидающий процесс добавляется к очереди активных процессов и производится начальная установка канала в пустое состояние.

Когда сообщение пересылается по внешнему каналу, процессор возлагает эту передачу на автономный интерфейс канала и исключает процесс из плана передач. Если интерфейс канала реализует передачу сообщения посредством прямого доступа к памяти, процессор вновь вводит ожидающий процесс в расписание пересылок. В то время когда имеет место внешняя пересылка сообщения, процессор может выполнять другие процессы.

Интерфейс включает три регистра, которые содержат:

- указатель рабочего пространства процесса;
- указатель сообщений;
- количество байтов подлежащих пересылке.

При необходимости ввода или вывода сообщения производится инициализация этих трех регистров и занесение указателя команды в рабочее пространство процесса. После того как оба процесса инициализируют свои каналные интерфейсы, происходит перепись сообщений и оба интерфейса ставят процесс в конец локальной очереди активных процессов.

Способ, последствием которого транспьютерное семейство реализует параллельную обработку при использовании нескольких транспьютерных СБИС, определяется организацией последовательных каналов и Оккам-каналов. Каждый последовательный канал позволяет реализовать два Оккам-канала, по одному для каж-

дого из двух направлений передачи. Пока ЦПУ занят обработкой, обмен информацией может происходить по всем четырем каналам.

При времени цикла транспьютера T414 50 нс на 4-е канала одновременно работающих в обоих направлениях со скоростью 100Мбит/с с использованием внутренней памяти затрагивается 8% производительности ЦПУ.

Сообщения пересылаются в виде последовательностей байтов. Каждый 8-битовый байт предваряется двумя единичными битами и сопровождается одним нулевым битом. В передающий транспьютер передается сигнал подтверждения приема состоящий из одного единичного и одного нулевого битов. Для обеспечения непрерывности передачи информации сигнал подтверждения начинается передаваться после получения двух заголовочных байтов.

Необходимо отметить, что архитектура транспьютера включает решение проблемы соединений. Для решения этой проблемы INMOS мог бы добавить несколько дополнительных коммуникационных каналов в чип, но общее число каналов оставалось бы все-таки небольшим ввиду ограничений технологии VLSI. Вместо этого INMOS избрал радикальное решение - добавление аппаратного мультиплексора, который позволяет четко делить физический коммуникационный канал.

В пару к чипу INMOS также разработал соответствующий высокопроизводительный чип коммутации, так что транспьютеры могут соединяться в полнодоступную сеть коммутации пакетов. Каналы связи между транспьютерами станут виртуальными каналами и их число определяется решаемой программой. С помощью чипов коммуникации, отвечающих за эффективную доставку сообщений по указанному назначению, виртуальные каналы могут запускаться между транспьютерами, которые не соединены напрямую физическими коммуникационными каналами.

Когда осуществляется программирование какой-либо задачи, то необходимо однозначно определить, какой физический коммуникационный канал закрепляется за каждым каналом передачи сообщений и может потребоваться несколько таких однозначных определений для приема сообщения от удаленного транспьютера. Это делает программу зависимой от точной тополо-

гии конкретной сети, следовательно, она становится неприложимой к сети с любой другой топологией.

В чипе два процесса, расположенные в произвольных местах сети. Они могут использовать один и тот же программный канал для связи. Аппаратура же сама упорядочит маршрут передачи сообщения. С этим улучшением, программы могут быть независимыми от топологии сети, и поэтому более переносимыми.

Архитектура транспьютера включает четыре физических коммуникационных канала. Но она содержит встроенный в чип контроллер коммуникаций, который поддерживает множество виртуальных каналов путем разделения сообщений в пакеты и поочередной передачи пакетов от нескольких сообщений через один физический коммуникационный канал.

Физические коммуникационные каналы чипа почти впятеро быстрее (100 Мбит/сек полный дуплекс), чем транспьютерные физические коммуникационные каналы. Кроме того, системный пакет обеспечивает более эффективное использование канала, поэтому производительность коммуникационного канала будет улучшена несмотря на сложность мультиплексирования.

Чип разделяет сообщения произвольной длины на последовательность 32-байтовых пакетов, в отличие от существующего транспьютера, который посылает сообщения байт за байтом. Смешанные пакеты, принадлежащие к различным сообщениям (т.е. к различным виртуальным каналам) требуют, чтобы каждый пакет имел заголовок, который бы говорил, какой виртуальный канал он использует.

Каждый пакет заканчивается специальным маркером EOP (End-Of-Packet - конец пакета), за исключением последнего пакета, который имеет маркер EOM (End-Of-Message - конец сообщения). Таким образом, нет нужды использовать дополнительную логику для счета пакетов, аппаратура знает, когда сообщение закончилось и позволяет передавать сообщения короче 32 байт (и последние части длинных сообщений, которые могут не занимать целый пакет).

Для сохранения синхронизации передачи сообщений, приемник должен подтверждать получение каждого пакета; подтверждение пакета - это пакет, не содержащий данных. Каждое вирту-

альное соединение содержит два виртуальных канала, один для передачи пакетов данных вовне, другой для приема подтверждений. Процесс посылки пакетов сообщений не может продолжаться до тех пор, пока передддача последнего пакета не будет подтверждена. Подтверждение посылается тогда, когда принимается первый байт пакета, оно разрешает непрерывную передачу сообщений, если приемник готов к вводу информации.

Запросы на передачу ставятся в очередь по каждому физическому коммуникационному каналу связи, так что от МП не требуется ожидания окончания передачи пакета.

В случае, если приемный процесс не готов к приему пакета, чип обеспечивает буферизацию ровно одного пакета на каждый физический коммуникационрый канал. По этой же причине существующие транспьютеры буферизуют 1 байт в аппаратном регистре. В чипе процесс буферизации будет осуществляться в память, что предпочтительнее буферизации в регистр, поэтому в случае достаточного объема и производительности памяти на одном процессоре может быть образовано любое число виртуальных каналов.

Для полной реализации концепции виртуальных каналов, требуется дополнительный чип переключения маршрутов С104 (коммутатор) (рис. 49).

Чип С104 является полным чипом обмена коммутируемыми пакетами, аналогичным по принципу работы телефонной сети органического доступа для обмена деловой информацией. Чип С104 содержит 32 физических коммуникационных транспьютерных канала и может коммутировать сообщение от одного из 32 чипов к любым другим чипам.

Чип С104 содержит коммутатор 32×32 для соединения любого из его физических коммуникационных каналов с любым другим, и чудесно простую логику для подключения адресата для каждого принятого сообщения. Чипу не требуется внутреннее CPU или значительное количество памяти благодаря схеме коммутации, примененной разработчиками INMOSa.

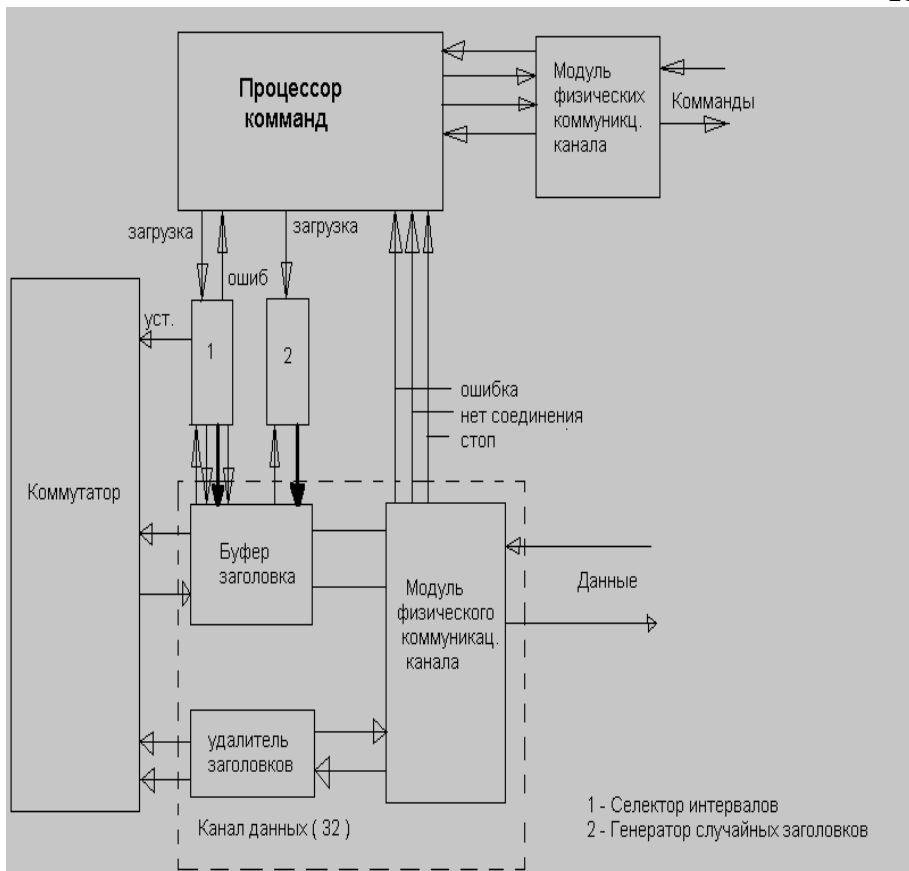


Рис.49

В большинстве сетей коммутации пакетов каждый переключатель маршрута принимает полный пакет и сохраняет его в буфере, декодирует адресную информацию пакета, и пересылает пакет к следующему узлу назначения. Эта схема не очень подходит для высокопроизводительных компьютерных сетей, поскольку она вносит относительно большую задержку между передачей пакета и его прибытием в пункт назначения (т.е. имеет большое время ожидания), к тому же требует от чипа коммутации C104 наличия промежуточной буферной памяти.

INMOS выбрал вместо этой схемы решение, называемое червячным каналом маршрутизации, в котором читается только

заголовок пакета и дешифруется его адрес. Затем, если требуемый физический коммуникационный канал свободен, остальная часть пакета посылается напрямую от входа к выходу потоком, без всякой буферизации.

Это означает, что заголовок пакета должен попасть в новый узел коммутации в то время, как тело пакета еще проходит через предыдущие узлы коммутации. В действительности, в сетях среднего размера, заголовок пакета может вообще не появиться от передающего абонента полностью, он достигнет пункта назначения еще до того, как абонент завершит его передачу. В результате, открывается сквозной канал от передающего абонента к принимающему, пока данные передаются непрерывно, и который закрывается, как только конец пакета "протянется" через канал. Аналогия с червячным каналом была выбрана из наблюдения за червяком, прокладывающим канал в песке, который закрывается снова за его хвостом.

В каждый момент времени, через промежуточный узел маршрутизации червячного канала может проходить только один пакет, поэтому любые другие пакеты, которым требуется этот узел, будут ожидать, пока узел не очистится. Итак, давая хорошую стратегию маршрутизации, червячный канал относится к каналам с очень малым временем ожидания, и он подразумевает, что чипу C104 нужен только 1-2 байтовый буфер для заголовков вместо буфера под целый пакет.

Червячный канал маршрутизации в общем невидим для сообщений посылающих и принимающих абонентов, поскольку он создается на аппаратном уровне ниже механизма синхронизации отправки/подтверждения.

Полная маршрутная стратегия требует решения алгоритма маршрутизации пакета, опираясь на адрес, содержащийся в его заголовке, т.е. какой канал будет открыт для передачи пакета. Алгоритмы маршрутизации являются хорошо изученной областью благодаря их колоссальной важности для индустрии передачи сообщений. Хороший алгоритм маршрутизации должен быть завершённым (т.е. он должен гарантировать, что каждый переданный пакет в результате достигнет абонента), не содержать в себе взаимоблокировок (dead-lock), должен быть оптимальным (т.е. пакеты

должны следовать по кратчайшим маршрутам) и малоизбыточным (т.е. пакеты должны иметь короткие заголовки).

Он также должен быть масштабируемым (т.е. хорошим для сетей любого размера) и многосторонним (т.е. хорошим для любой топологии сети). Вдобавок, если он предназначен для работы на недорогом, быстром оборудовании, хороший алгоритм маршрутизации должен быть простым. Простота вдвойне важна, когда вы используете быстрый маршрутизатор червячного канала, во избежание ухудшения времени ожидания сети. Не существует алгоритмов, полностью удовлетворяющих всем этим требованиям.

Одним стандартным решением является таблица маршрутизации, которая включает в себя в виде списка все узлы сети и все их соединения, подобно телефонному справочнику; этот алгоритм является завершенным, оптимальным и многосторонним, но не дешевым, быстрым или масштабируемым. Другим общим решением является побитное разрушение, в котором каждый узел сети, которого достигает сообщение, "глочет" один бит из заголовка сообщения; этот алгоритм является завершенным, дешевым, быстрым и оптимальным, но не многосторонним или масштабируемым, поэтому он предназначается для структуры типа двоичное дерево фиксированного размера.

INMOS избрал сравнительно новый алгоритм, называемый интервальной маршрутизацией (рис.50), который является завершенным, свободным от взаимоблокировок, недорогим, быстрым и масштабируемым, могущим быть близким к оптимальному или многосторонним, в зависимости от того, как он используется. Каждый транспьютер (или другой узел назначения, такой, как межсетевой шлюз или периферийный чип) маркируется номером, так сеть из N транспьютеров будет маркирована $0,1,2... N-1$. Этот номер используется как адрес назначения в заголовках пакетов.

В каждом переключателе маршрута каждый физический коммуникационный канал маркируется интервалом возможных величин заголовка, и пакеты передаются через тот канал, интервал которого попадают величины их заголовков.

Описание интервала $(0,3)$ должно интерпретироваться как указание на то, что величина заголовка пакета должна больше или равна 0 и меньше 3, для того, чтобы лежать в этом интервале. Ин-

тервалы не перекрываются и пронумерованы так, что каждый заголовок попадает только в один интервал.

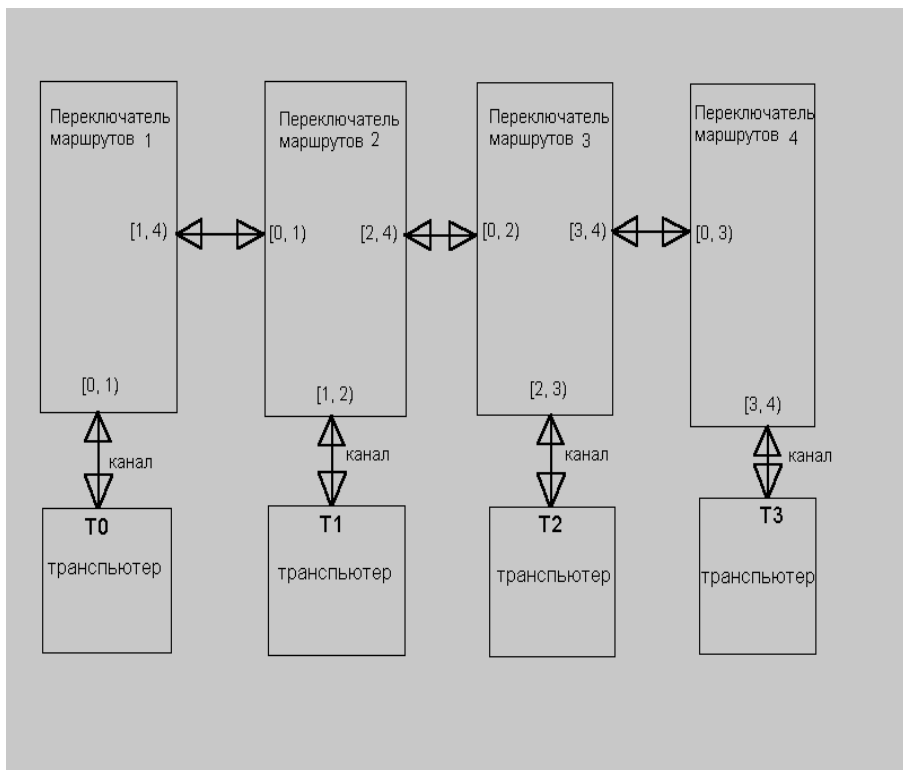


Рис. 50

В результате каждый интервал говорит: "В этом направлении лежит часть сети, содержащая все указанные в интервале номера процессоров". Для разработчика аппаратуры привлекательность метода интервальной маршрутизации заключается в том, что он может быть применен для маршрутизации пакета с использованием одной единственной операции сравнения. Так логика чипа S104 состоит чуть больше, чем из пары компараторов и интегральных регистров на каждый физический коммуникационный канал. Чип S104 может использовать одноили двухбайтовые заголовки, поэтому в одиночной сети может быть маршрутизировано до 65536 транспьютеров.

Хотя схема интервальной маршрутизации может быть гарантирована от взаимоблокировок, она может стать жертвой горячих точек. Когда слишком много сообщений маршрутизировано через один узел сети, большинство доставляемых пакетов остановится на непредсказуемый интервал времени, решающим образом снижая скорость прохождения через узел, такой узел сети называется горячей точкой. Эта проблема сильно отличается от взаимоблокировки, которая является логической ошибкой и результат которой бесконечное ожидание сообщения, это скорее свойство топологии сети и алгоритма маршрутизации, чем применения. Для любой топологии сети, как вы можете догадаться, вероятно найдется алгоритм применения, который может породить горячие точки.

Можно избежать горячих точек равномерным распределением сетевого трафика, и можно запрограммировать чип C104 для выполнения этого распределения. Маршрутизация становится двухфазным процессом, в первой фазе которого пакет посылается в случайно выбранный узел сети, а из этого узла сообщение посылается к его пункту конечного назначения - схема называется универсальной маршрутизацией.

Очевидно, что универсальная маршрутизация увеличит время ожидания в сети и ограничит ее максимальную пропускную способность, но использование этого алгоритма гарантирует, что наихудшая производительность сети будет не сильно отличаться от максимальной. С горячими точками наихудшая производительность сети может ухудшиться на порядки, и фактор дешевизны сети перестает играть роль вследствие непригодности сети для работы. Физически схема универсальной маршрутизации добавляет дополнительный случайный заголовок в начале каждого пакета, который удаляется снова внутри случайного промежуточного узла сети.

Несмотря на то, что механизм виртуальных каналов вызывает наибольший интерес, процессор содержит ряд других улучшений для облегчения жизни разработчиков операционной системы и для улучшения поддержки языков, таких как C и АДА.

Подобно существующему T800, чип будет иметь встроенный сопроцессор плавающей точки (FPU) и четыре физических

коммуникационных канала. Ядро процессора будет иметь встроенное кеширование и контроллер DRAM, который будет поддерживать адресные режимы статической памяти. Эти возможности в сочетании с улучшенной технологией изготовления дают перспективу достижения пиковой производительности в 100 млн. операций в секунду и 20 млн. операций с плавающей точкой в секунду. INMOS планирует объединить схему защиты памяти, что позволит каждому процессору иметь доступ к четырём областям памяти. Эти защищённые области могут быть использованы для программ, данных, стека и динамической области памяти при применении операционных систем типа UNIX.

8.6.7. MBC – 1000

MBC-1000 — система 3-го поколения, разработанная в России фирмой «Квант». Она основана на использовании микропроцессоров Alpha 21164 (разработка фирмы DEC-Compaq; выпускается также заводами фирм Intel и Samsung) с производительностью до 1-2 млрд. операций в секунду и присоединенной оперативной памятью объемом 0,1-2 Гбайт.

Мультипроцессорный массив системы с блоками вторичного электросилового питания и вентиляцией располагается в стойках размером 550x650x2200 мм³ промышленного стандарта; вес заполненной стойки — 220 кг, потребляемая мощность до 4 кВт. Система MBC-1000 с производительностью до 1 TFLOPS состоит из 8 стоек (512 узлов).

Система реализуется в двух модификациях: на базе «транспьютероподобного» связанного микропроцессора TMS320C44 (фирма Texas Instruments), имеющего 4 канала с пропускной способностью каждого — 20 Мбайт/с, либо на базе связанного микропроцессора SHARC ADSP 21060 (фирма Analog Devices), имеющего 6 внешних каналов с пропускной способностью каждого — 40 Мбайт/с.

Исполнение MBC-1000К отличается использованием для межпроцессорного обмена коммутационной сети MYRINET (фирма Murgisom, США) с пропускной способностью канала в дуплексном режиме 2x160 Мбайт/с. Кроме того, предусмотрено подклю-

чение к каждому процессору памяти на жестком диске с объемом 2-9 Гбайт. В стандартной стойке располагается до 64 процессоров системы МВС-1000 или 24 процессоров системы МВС-1000К. Предусмотрены средства системного объединения стоек для установок с большим числом процессоров.

Процессорные узлы связаны между собой по оригинальной схеме, сходной с топологией двумерного тора (для 4-линковых узлов). Структурный модуль (рис.51) состоит из 16 вычислительных модулей (ВМ), образующих матрицу 4x4. При этом четыре угловых элемента матрицы соединяются через транспьютерные линки по диагонали попарно. Оставшиеся 12 линков предназначаются для подсоединения внешних устройств (4 линка угловых ВМ) и соединений с подобными модулями. Максимальная длина пути в таком структурном модуле равна трем (против шести в исходной матрице 4x4).

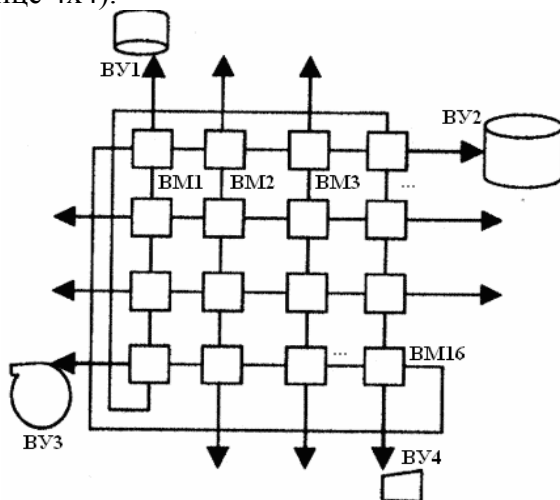


Рис.51

Конструктивным образованием МВС-1000 является базовый вычислительный блок, содержащий 32 вычислительных модуля (рис.52). Максимальная длина пути между любыми из 32 вычислительных модулей равна пяти, как в булевском гиперкубе. При этом число свободных линков после комплектации блока составляет 16, что позволяет продолжить проце-

дуру объединения. Возможна схема объединения двух базовых блоков в 64-процессорную систему, которая приведена на рис. 53.

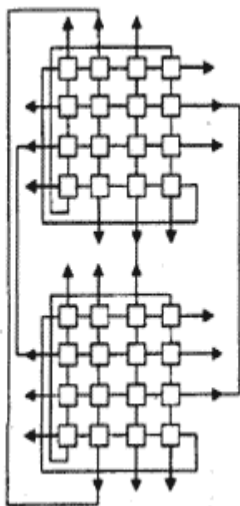


Рис. 52

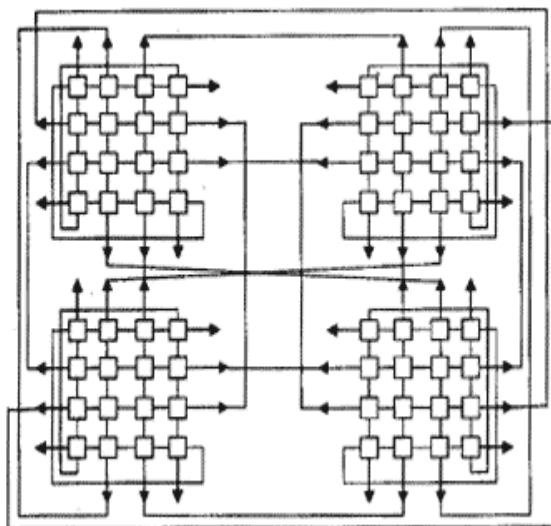


Рис.53

Для управления массивом процессоров и внешними устройствами, а также для доступа к системе извне используется так называемый хост-компьютер (управляющая машина). Обычно это рабочая станция AlphaStation с процессором Alpha и операционной системой Digital Unix (Tru64 Unix) или ПК на базе Intel с операционной системой Linux.

Начиная с 1999 года, все вновь выпускаемые МВС-1000 строятся как кластеры выделенных рабочих станций. Это означает, что, в отличие от ранних версий МВС-1000, в качестве вычислительного модуля используются не специализированные ЭВМ, предназначенные только для применения в качестве деталей суперкомпьютерной установки, а обычные, универсальные персональные компьютеры. Соответственно, и в качестве коммуникационной аппаратуры используются не специализированные «транспьютероподобные» процессоры, а обычные сетевые платы и коммутаторы, применяемые для построения офисных локальных сетей. Такой подход стал не только возможным, но и единственно

оправданным, по мере совершенствования коммуникационной аппаратуры общего назначения, в первую очередь, с появлением современных сетевых коммутаторов.

В качестве базовой ОС узла используется Linux, что является, фактически, общепринятым мировым стандартом для построения систем такого класса. Это позволило многократно расширить и упростить, по сравнению с ранними версиями MVS-1000, адаптацию самого разнообразного программного обеспечения, как свободно распространяемого, так и коммерческого.

8.6.8. Молекулярные компьютеры.

Ученые из Калифорнийского университета в Лос-Анджелесе, работающие в составе группы Hewlett-Packard, заявили о создании молекулярных переключателей, которые способны функционировать как элементы памяти.

Над созданием систем нового поколения исследовательские группы таких крупных корпораций, как IBM и Hewlett-Packard, работают уже давно. Hewlett-Packard специализируется на создании молекулярных, а IBM - квантовых компьютеров.

Преимущества молекулярных компьютеров состоят в чрезвычайно малых размерах и в мизерном количестве потребляемой энергии. В отличие от обычных компьютеров, в молекулярных, вместо кремниевых транзисторов, используются молекулы. Ранее та же группа исследователей смогла заставить молекулы ротаксана переходить из одного состояния в другое. Таким образом, можно было реализовать постоянное запоминающее устройство, вроде CD-ROM диска.

Использование молекулярных элементов для записи информации открывает новые физические возможности повышения эффективности вычислительных систем. Это объясняется существенным уменьшением потерь энергии на выполнение одной логической операции, массовым параллелизмом, высокой плотностью записи информации вплоть до 1 бита на несколько атомов.

Существует огромное число молекулярных систем, имеющих несколько устойчивых состояний (электрических, магнитных, оптических, химических), в которых можно осуществлять перехо-

ды между этими состояниями под действием внешних факторов (электрическое и магнитное поля, электромагнитное излучение, химические реакции). Из таких молекул или групп молекул могут быть сконструированы различные элементарные вычислительные устройства. Размеры этих элементов составляют 10-50 нм и, следовательно, позволяют создавать устройства с плотностью записи до 10^{12} бит/см². При этом, быстродействие молекулярных элементов определяется временами электронных переходов и может достигать величин порядка 10^9 - 10^{12} оп/сек. Энергопотребление внутримолекулярных процессов очень мало и составляет 10^{-8} - 10^{-10} Вт.

Перечисленные теоретически возможные преимущества молекулярных систем известны давно, однако не было методов создания упорядоченных молекулярных ансамблей. Одним из первых методов решения этой задачи была технология пленок Ленгмюра-Блоджетт.

Первые двоичные молекулярные элементы были основаны на изменении состояния молекулярного ансамбля при воздействии на него электрического сигнала. Одним из таких элементов, является молекулярная цепочка, представляющая собой чередование потенциальных ям и барьеров. В такой цепочке возможен эффект резонансного туннелирования. При совпадении энергии входящего в цепочку свободного электрона с одним из уровней в потенциальной яме вероятность прохождения электрона через цепочку равна 1. При подаче потенциала на одну из молекул, соответствующих квантовой яме уровни смещаются, условия резонанса нарушаются и цепочка становится непроводящей. На основе эффекта резонансного туннелирования можно реализовать различные логические элементы. Для создания таких элементов используются молекулы порфиринов, соединенные углеводородными мостиками.

Для управления состоянием отдельной молекулы применяют зонд туннельного микроскопа, который можно использовать для исследований при разработке и создании молекулярного компьютера.

Архитектура параллельных вычислительных систем, традиционно используемая в разрабатываемой вычислительной технике, реализуется на основе твердотельной микроэлектронной

элементной базы, но совершенно очевидно, что такая архитектура вполне подходит для молекулярных систем. Идея параллельных вычислений открывает путь к решению двух принципиальных проблем создания молекулярных вычислительных устройств. С одной стороны решается проблема адресации, т.к. отпадает необходимость контакта к каждому логическому молекулярному элементу. С другой стороны, многие типы параллельных вычислительных систем устойчивы к выходу из строя определенного числа логических элементов. Это происходит в результате технологических дефектов молекулярных слоев, которые не приводят к выходу из строя всей вычислительной системы, а лишь ослабляют интенсивность выходного сигнала.

В качестве вычислительных систем параллельного действия представляется перспективным использование таких природных биологических объектов, как бактериородопсин, родопсин, фитохром, ряд ферментов и белков, а также природные нейроструктуры. При использовании молекулярных структур необходимо решить такую очень важную проблему, как стыковка биологических молекулярных объектов с макроскопическими опто- и микроэлектронными устройствами ввода-вывода информации. В качестве основного микроэлектронного объекта стыковки элементов памяти с устройствами управления можно рассматривать полевой МДП-транзистор, затвор которого связан с молекулярным фрагментом памяти. Очевидно, что при площади затвора транзистора 1 мкм^2 будет восприниматься суммарный отклик от большого числа молекул (10^3 - 10^4).

Проблему по снятию сигнала с такого количества молекул можно решать различными методами. Одним из таких методов является стыковка обычных МДП полевых транзисторов с фрагментами пурпурных мембран, которые представляют собой белковые молекулы бактериородопсина, упакованные в двухслойную липидную мембрану. Основным свойством молекулы бактериородопсина является перенос протона с одной стороны мембраны на другую при поглощении кванта света с длиной волны 570 нм. Акт переноса протона может быть зарегистрирован с помощью полевого транзистора, в котором на месте затвора расположена пурпурная мембрана.

Поскольку линейные размеры пурпурных мембран сравнимы с размерами транзисторов (до 1,0 мкм), то такая стыковка оказывается вполне реальной. Слои бактериородопсина, состоящие из пурпурных мембран могут быть нанесены на подложку методом Ленгмюра-Блоджетт. Таким образом, можно выполнять стыковку микроэлектронных систем с молекулярными, а также производить преобразование оптических сигналов в электрические.

Основной особенностью перечисленных выше молекулярных носителей, используемой разработчиками запоминающих структур, является наличие двух состояний молекулярной структуры. Указанная способность позволяет определять текущее состояние молекулы с помощью лазера, настроенного на соответствующую частоту.

Рассмотрим систему памяти, в которой молекулярный носитель запоминает данные в трехмерной матрице в виде прозрачной капсулы, заполненной полиакридным гелем. Один из вариантов такой капсулы может иметь размеры 20 x 20 x 40 мм. Для функционирования памяти капсулу помещают в запоминающую структуру, состоящую из нескольких лазеров и детекторной матрицы, реализованной на базе прибора, использующего принцип зарядовой инъекции, и позволяющей производить запись и чтение данных.

Перед записью данных с помощью желтого страничного лазера на 590 нм производится активизация возбужденной плоскости в материале внутри капсулы. Получение энергоактивной плоскости в виде страницы данных, требует наличия пространственного светового модулятора. Таким модулятором может быть жидкокристаллическая матрица, создающая маску на пути лазерного луча. Таким образом, при активизации записывающей плоскости происходит перевод молекул в 0 - состояние, соответствующее значению бита "0".

Для записи информации зажигается красный лазер, перед которым также устанавливается пространственный световой модулятор, отображающий двоичные данные. При этом последний создает на пути луча необходимую маску, способствующую облучению только определенных точек страницы. Молекулы в таких

местах будут соответствовать двоичной единице в отличие от оставшейся части страницы, представляющей двоичные нули.

Для чтения информации необходимо включить считывающий лазер, при этом, из-за различия в спектрах поглощения, можно идентифицировать двоичные нули и единицы. Молекулы, представляющие двоичный ноль, поглощают красный свет, а представляющие двоичную единицу пропускают луч мимо себя. В результате получается рисунок из светлых и темных участков на фоторегистрирующей матрице. Последняя позволяет хранить страницу цифровой информации.

Для стирания данных достаточно короткого импульса синего лазера, чтобы вернуть молекулы из состояния двоичной единицы в двоичный ноль. Страница данных может быть прочитана без разрушения до 5000 раз, после чего освежается с помощью новой записи.

С помощью предложенной запоминающей структуры можно получить доступ к данным со скоростью до 90 МВ/с при условии объединения по восемь запоминающих битовых ячеек в байт с параллельным доступом и соответствующей схмотехнической реализации системы памяти. Емкость данных в рассмотренной кювете может достигать до 1,4 GB, поскольку дальнейшее увеличение емкости данных связано с конструктивными особенностями линзовой системы, а также с качеством непосредственно самого молекулярного носителя.

При обработке молекулярного носителя лазерным лучом во время записи-стирания с энергией выше 8- 10 мДж/см² не достигается существенного увеличения уровня сигнала считывания информации. Сигнал с фоторегистрирующей матрицы обрабатывается и регистрируется на ЭВМ.

Проведенные исследования позволяют говорить о хранении и обработке информации на молекулярном уровне. Идентификация двоичных нулей и единиц производится за счет разности спектров поглощений лазерного излучения различными состояниями исследуемой молекулярной структуры. При этом молекулы, поглощающие свет, отвечают двоичному нулю, а молекулы, пропускающие лазерный луч мимо себя, отвечают двоичной единице.

Исследуемые молекулярные запоминающие структуры,

кроме сверхвысокой плотности записи информации, почти не имеют потерь энергии на выполнение одной логической операции, а также позволяют достигать массового параллелизма. Гибридные системы хранения и обработки информации, которые совмещают свойства твердотельных ЭВМ и молекулярных вычислительных систем, могут быть эффективно использованы в различных отраслях науки и техники.

Значительные перспективы, которые открывает технология записи и считывания информации на светочувствительном материале, позволяют обеспечить максимальную скорость доступа к данным. Это достигается за счет того, что обрабатываемый массив данных кодируется в один большой блок данных, который, в свою очередь, записывается всего за одно обращение. Если учесть, что такая запоминающая система для обработки информации не содержит движущихся частей и доступ к страницам осуществляется параллельно, то запоминающее устройство в целом может позволить хранить информацию объемом в десятки гигабайт.

Молекулярные компьютеры, по мнению специалистов, должны быть реализованы путем объединения кремниевого чипа с молекулярными компонентами в один гибридный чип. Такой чип будет обладать в тысячи раз большим быстродействием, а память построенная на таком принципе может быть в миллионы раз объемнее. И главное, что подобные технологии будут крайне дешевы, в десятки раз дешевле современных технологий.

Сама идея этих логических элементов не является революционной: кремниевые микросхемы содержат миллиарды точно таких же. Но преимущества в потребляемой энергии и размерах способны сделать компьютеры вездесущими. Молекулярный компьютер размером с песчинку может содержать миллиарды молекул. А если научиться делать компьютеры не трехслойными, а трехмерными, преодолев ограничения процесса плоской литографии, применяемого для изготовления микропроцессоров сегодня, преимущества станут еще больше.

Кроме того, молекулярные технологии сулят появление микромашин, способных перемещаться и прилагать усилие. Причем для создания таких устройств можно применять даже традиционные технологии травления. Когда-нибудь эти микромашинки

будут самостоятельно заниматься сборкой компонентов молекулярного или атомного размера.

Первые опыты с молекулярными устройствами еще не гарантируют появления таких компьютеров, однако это именно тот путь, который предначертан всей историей предыдущих достижений. Массовое производство действующего молекулярного компьютера вполне может начаться где-нибудь между 2005 и 2015 годами.

8.6.9. Оптические МПС

Оптические системы в качестве носителя информации используют свет. Простейшим способом передачи данных при помощи луча света является его модуляция по принципу "есть свет - нет света". У света есть одна особенность: распространяясь в пространстве, он всегда заполняет собой какую-то его часть. Следовательно, существует возможность кодирования информации путем изменения амплитуды и фазы луча не только во времени, но и в пространстве.

Формой реализации этой идеи является оптический элемент, называемый транспарантом. Попросту говоря, это кусочек прозрачного материала, на который каким-либо способом нанесено изображение, представляющее собой пространственное (в координатах, связанных с транспарантом) распределение коэффициента поглощения, коэффициента преломления (или толщины) или же того и другого одновременно. В первом случае получим транспарант, модулирующий пространственное распределение амплитуды проходящего через него света. Примером такого транспаранта может являться кадрик обычной фотопленки.

Во втором - получаем так называемый фазовый транспарант. Как правило, он совершенно прозрачен, и невооруженным глазом невозможно разглядеть на нем никакого изображения. Материал пластинки, будучи неоднороден по величине коэффициента преломления или, чаще всего, по толщине, изменяет (модулирует) распределение фаз проходящей сквозь него световой волны - по той простой причине, что различные участки такого транспаранта пропускают свет за разное время. В конечном итоге это приводит к

возникновению за пластинкой транспаранта заданного пространственного распределения фаз. Еще говорят о формировании определенного фазового профиля волнового фронта (рис. 54).

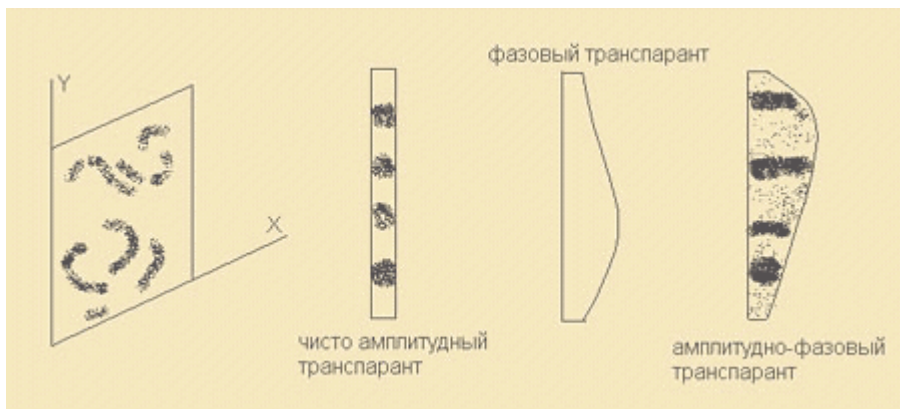


рис. 54

Из сказанного выше становится очевидным, что любой самый привычный оптический элемент - будь то линза или целый объектив - представляет собой амплитудно-фазовый транспарант (иногда используют термин "фильтр").

Транспаранты можно изготавливать на базе технологий, позволяющих управлять оптическими свойствами вещества (жидкие кристаллы или электрооптические материалы), что дает возможность помимо пространственной модуляции осуществлять еще и временную, а также строить адаптивные, в том числе с обратными связями оптические системы.

В технологии оптической обработки данных информацию удобно представлять пространственно-временным распределением амплитуды и фазы светового луча.

Когда мы говорим об амплитудно-фазовой модуляции света, мы, по существу, уже касаемся вопросов не только кодирования или представления информации, но и ее обработки.

Само понятие "модуляция" ничего не говорит о том, что происходит с модулированным сигналом дальше. Привычно думать, что ничего особенного с ним не происходит. Распространяется себе и все.

В данном случае в пространстве за транспарантом световая волна не "просто распространяется". Там, в этой области, интерферируют части волны, прошедшие различные участки транспаранта (в общем случае по принципу "каждый - со всеми"), и формируется новая структура волны, порой совсем не похожая на первоначальную. И этот процесс является управляемым.

Именно поэтому здесь уместно говорить не столько о модуляции, сколько о преобразовании структуры волны, то есть, по сути, об обработке информации, записанной в этой структуре.

Сейчас мы затронули чрезвычайно важный, я бы сказал, принципиально важный момент. Из математики мы знаем, что любые процессы или объекты могут быть представлены как сумма (ряд) некоторых элементарных периодических функций, например, гармонических колебаний. Этот ряд называют еще спектром или спектральным представлением объекта.

Любые изменения некоторого объекта, допустим, оптического изображения, связаны с изменениями его спектрального состава. Поэтому транспаранты в технике оптических вычислений называют еще пространственными амплитудно-фазовыми фильтрами, подчеркивая, что они осуществляют некоторый процесс фильтрации, то есть изменения состава пространственных частот исходного изображения. Именно поэтому и говорят, что комбинация пространственного фильтра и некоторого объема пространства за ним - уже представляет собой оптический процессор.

Как это все эти принципы работают "на практике". Создадим экспериментальную установку: ширма с тремя источниками света А, В и С, экран (см. рис. 55).

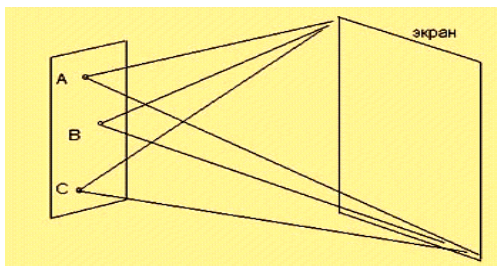


рис. 55

Как сделать, чтобы на экран попадал свет только от источников А и С?

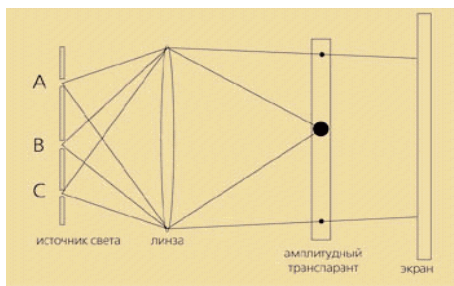


рис. 56

Поставим на пути света линзу (фазовый транспарант). В пространстве за собой она построит изображения источников так, как это показано на рис. 56. Если теперь дополнить нашу установку амплитудным транспарантом-фильтром (попросту говоря, стеклянной пластинкой с непрозрачным участком напротив изображения источника В), то в пространстве за фильтром будет по-прежнему присутствовать свет от источников А и С, но света от источника В там не будет.

В случае использования в системе когерентного света при помощи описанного устройства можно фильтровать изображения источников света и световое поле объектов произвольной конфигурации, то есть даже в тех случаях, когда ими являются просто физические объекты.

Три объекта А, В и С освещены когерентным светом. На экран падает свет, отраженный всеми предметами. Вместо экрана, кстати, можно взять фотоаппарат. Тогда на пленке будут сфотографированы все три объекта.

Можно ли сделать так, чтобы фотоаппарат, снимая сцену, не зафиксировал объект В? Как "вычистить" отраженное от него световое поле из общего поля сцены? В рамках методов обычной оптики эта задача неразрешима.

Для решения этой задачи применяют специальный вид амплитудно-фазовых транспарантов, которые называются комплексно-сопряженными фильтрами. Математические подробности авто-

ры не приводят, но отмечают, что эти оптические элементы обладают свойством собирать в одну точку световое поле, обладающее некоторой заранее заданной структурой. Если нам известна конфигурация (амплитудно-фазовое распределение) поля, рассеянного объектом В, то, изготовив соответствующий фильтр, можно решить поставленную задачу, например, так (рис. 57):

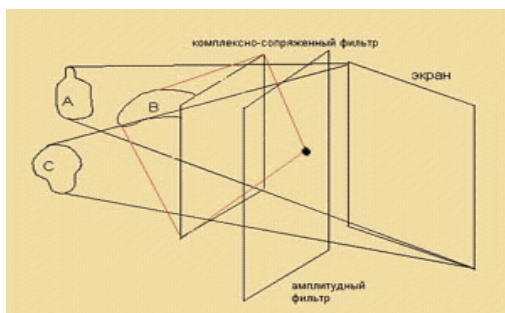


Рис.57

Свет, рассеянный объектом В, собирается в точку, где перекрывается амплитудным фильтром. Теперь на пленке в фотоаппарате будет формироваться все та же сцена, что и раньше, но объекта В на ней не будет.

Описанные процессы используются в качестве основы для проектирования оптических устройств распознавания образов или быстрого поиска необходимой информации в больших массивах данных, сформированных на оптических носителях - фото- или голографических транспарантах.

Теперь самое время рассмотреть случаи, когда требуется не просто выделение или распознавание требуемого информационного объекта, но именно преобразование его.

Из схемы записи (съемки) голограммы (рис. 58), понятно, что, во-первых, голограмма в сути своей представляет не что иное, как амплитудно-фазовый транспарант, на котором зафиксировано поле, получившееся в результате интерференции опорного пучка и отраженного от объекта света. А во-вторых, что опорный источник сам может являться некоторым объектом.

Тогда становится очевидным, что если обычная голограмма, освещаемая опорным лучом, восстанавливает изображение объекта съемки, то голограмма, полученная путем записи интерференции света от двух объектов, будет в свете одного из них восстанавливать изображение другого и наоборот (это свойство носит название "принцип обратимости голограмм").

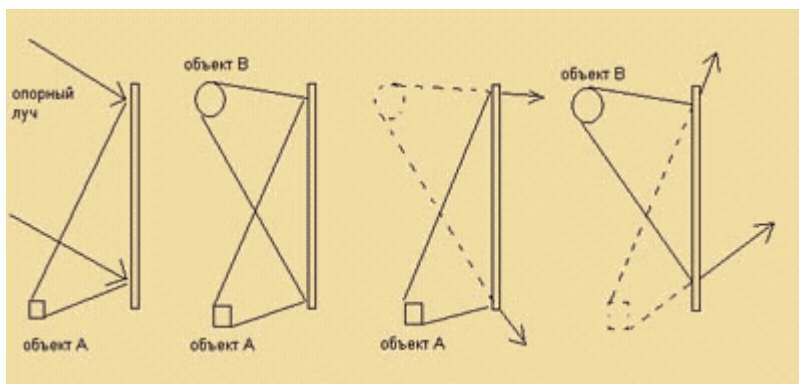


рис. 58

Практическая польза здесь очевидна: получается возможность осуществить "преобразование" любого изображения в заранее заданное другое (строго говоря, не преобразование, а как бы "замену" или "подстановку").

Много возможностей таит включение в схемы оптических вычислителей контуров обратных связей и нелинейных элементов!

Так же много перспектив сулит интенсивно развивающийся в последние годы подход к созданию оптических процессоров как квантовых устройств с распределенными параметрами. В них уже не найти привычных линз, зеркал или пластинок-фильтров. Их функции распределены по объему рабочего пространства процессора. Массовое производство такого рода устройств по своему революционному воздействию на отрасль сравнимо с началом выпуска интегральных микросхем.

Обработка информации в оптической системе может осуществляться как в процессе переноса изображения (представляющего собой специальным образом подготовленный входной сиг-

нал) через оптическую систему, реализующую вычислительную среду, так и путем осуществления переключений в так называемом оптическом транзисторе. Можно легко показать, что при линейных размерах изображения 1 см, разрешении 3 мкм и длине оптической системы порядка 30 см (давно доступные оптикам технологические нормы) можно получить пиковую производительность порядка 10^{16} элементарных операций в секунду!

Коммутация информационных каналов в оптическом компьютере осуществляется с большой скоростью и отличается простотой реализации за счет того, что лучи света в пустом пространстве распространяются, не взаимодействуя друг с другом. По сравнению с обычной электроникой выигрыш очень быстро растет с ростом числа коммутируемых каналов. Использование третьего измерения для ввода/вывода информации в оптоэлектронных чипах создает дополнительные возможности, которым у электронных соединений нет никаких аналогов.

Еще одно уникальное свойство оптических систем: в прозрачной среде информация, закодированная оптическим лучом, может обрабатываться без затрат энергии. Естественно, закон сохранения энергии при этом не нарушается. Наконец, отметим, что оптическая система ничего не излучает во внешнюю среду, обеспечивая защиту компьютера от перехвата информации. И обратно: оптическая система надежно защищена от сторонних электромагнитных наводок.

Основные элементы оптических компьютеров с переносом изображения давно известны. Это - линза, зеркало, оптический транспарант и слой пространства. В настоящее время к ним добавились волноводные элементы, а также лазеры, полупроводниковые многоэлементные фотоприемники, нелинейные оптические среды, разного рода дефлекторы и светоклапанные устройства.

Базисная логическая функция, элементарный кирпичик, с помощью которого можно построить любой, сколь угодно сложный цифровой компьютер, имеет множество оптических реализаций. На рис. 59 дан простой пример построения многовходовой функции ИЛИ-НЕ/И-НЕ с помощью линзы L и порогового устройства-инвертора N.

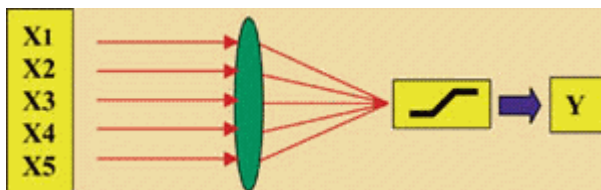


Рис. 59

Здесь в качестве порогового элемента можно использовать как оптическое светоклапанное устройство (переключающаяся, бистабильная оптическая среда), так и простой фотоэлектронный приемник с нелинейной передаточной характеристикой (то есть нелинейной зависимостью интенсивности выходного светового потока от входного).

На рис. 60 показан оптический процессор, реализующий произвольное матричное преобразование входного вектора-строки в выходной вектор-столбец. Здесь LED - линейка светоизлучающих диодов. Они расположены на фокальной линии цилиндрической линзы L1. T - оптический транспарант с записанной на нем матрицей пропускания $T(i, j)$.

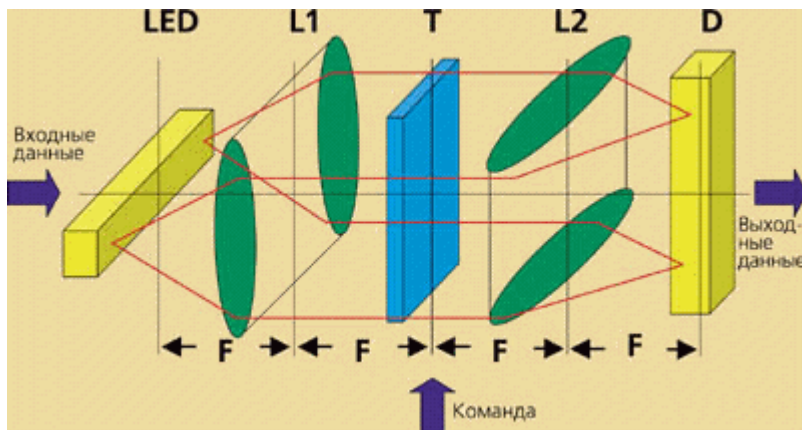


Рис. 60

Строки матрицы параллельны образующей первой линзы. L2 - цилиндрическая линза, образующая которой параллельна столбцам матрицы транспаранта. Она собирает лучи, прошедшие

через элементы одной строки, на одном пикселе многоэлементного линейного фотоприемника D. Нетрудно видеть, что входной X и выходной Y вектора связаны линейным преобразованием $Y=TX$

В оптической системе возможна также обработка двумерных структур. На рис. 61 представлена схема оптического процессора, реализующего операцию свертки двух изображений, которая лежит в основе работы многих устройств ассоциативной памяти и распознавания образов.

Здесь S - плоский однородный источник света, L1 и L2 - сферические линзы, D - матричный фотоприемник, T1 и T2 - транспаранты, пропускание которых соответствует двум обрабатываемым изображениям. Распределение интенсивности излучения на матричном фотоприемнике пропорционально интегралу $J(x, y) = \iint T1(x-u, y-v) T2(u, v) du dv$.

В предыдущих примерах свет выполнял ту же роль, что и электроны в проводниках обычных микросхем. При этом в качестве "проводов" выступали геометрические лучи. Понятно, что с таким же успехом свет можно загнать в волновод и организовать вычислительную среду по принципам, близким к идеологии электронной полупроводниковой микросхемотехники. Этим занимается интегральная и волоконная оптика.

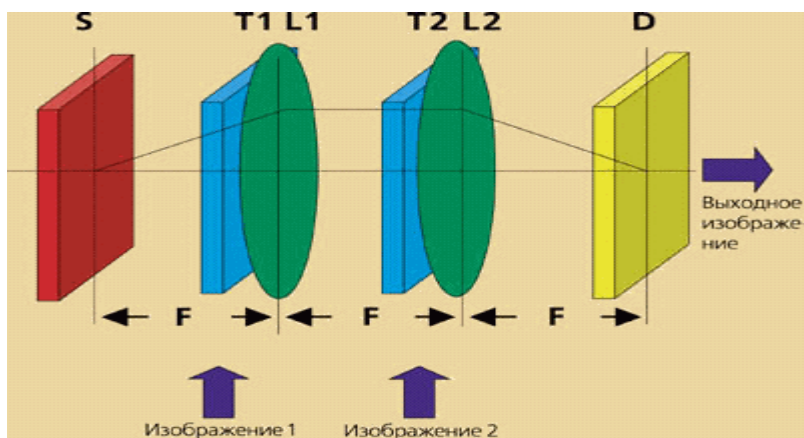


Рис. 61

Принципиально новые возможности дает использование свойств пространственной когерентности излучения. Структура когерентного оптического процессора, так называемая 4F-схема, приведена на рис. 62. Здесь LS - лазерная осветительная система, формирующая широкий пучок когерентного излучения. T1 и T2 - амплитудно-фазовые транспаранты, модулирующие фазу и амплитуду проходящей световой волны. L1 и L2 - сферические линзы с фокусным расстоянием F . Результирующий сигнал считывается матричным фотоприемником D.

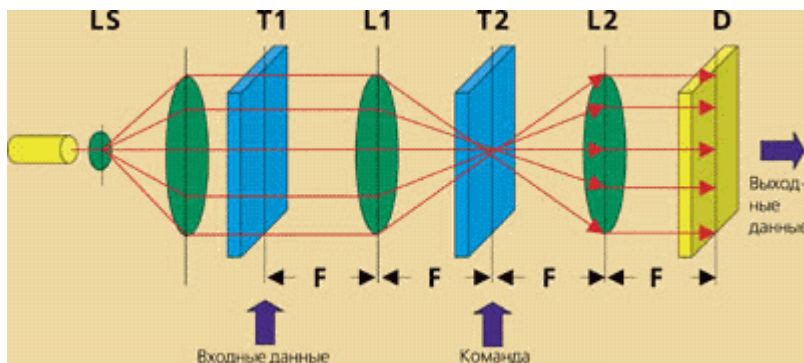


Рис. 62

Распределение амплитуды светового поля в плоскости фотодетектора пропорционально свертке амплитудного пропускания первого транспаранта с Фурье-образом амплитудного пропускания второго транспаранта. Процессоры такого типа используются в качестве комплексных пространственных фильтров в системах улучшения качества изображения, а также в системах распознавания образов.

Если же нас интересует Фурье-спектр двумерного сигнала, то он вычисляется с помощью линзы L и слоя пространства длиной F так, как показано на рис. 63. Остальные элементы предназначены для ввода-вывода данных и для освещения системы.

Отметим, что для обычного компьютера, использующего быстрый алгоритм Кули-Тьюки, длительность выполнения Фурье-преобразования растет с ростом числа точек дискретизации n пропорционально $n \log(n)$. В оптическом компьютере эта процедура даже в двумерном случае выполняется всего за один машинный

такт, что делает оптический компьютер незаменимым для военных целей, а также для решения задач, требующих быстрой оценки ситуации и управления в реальном времени.

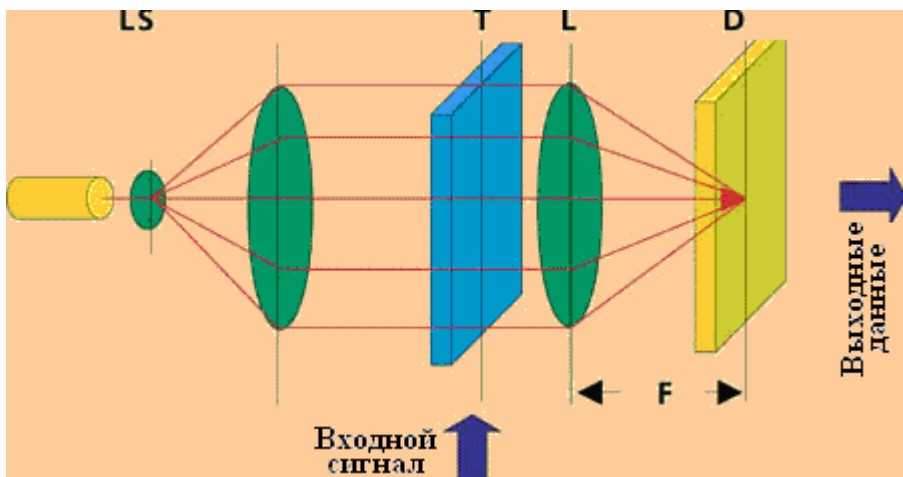


Рис. 63

Замечательным свойством оптического компьютера с передачей изображения является его способность за один такт обрабатывать двумерные картинки, причем машинная команда сама может представляться картинкой. Откуда же эти картинки брать? Их источником могут быть внешняя среда, результат предыдущего вычисления, оптическая память. Если из внешней среды изображение можно ввести с помощью объектива, а результат предыдущего вычисления возвратит на вход процессора с помощью системы зеркал, то в качестве памяти можно использовать различные оптические и оптоэлектронные устройства записи, хранения и извлечения изображений. Например, кассету со слайдами.

Особый интерес представляют голографические устройства памяти. Такая память обладает рядом достоинств. Голограмма сохраняет информацию не только об интенсивности, но и о фазе световой волны, что в оптике принципиально важно, а с утилитарной точки зрения - позволяет повысить объем записываемой информации. Кроме того, различные картинки можно записывать в одно и то же место, используя весь объем носителя, а не тонкий слой по-

верхности (как в случае обычной оптической или магнитной памяти). По оценкам специалистов, объемная плотность записи информации может превышать величину 10^{11} бит/см³, а скорость ввода информации с голограмм - несколько гигабит в секунду.

Кроме того, голограмма сама может использоваться в качестве принципиального узла оптического процессора, выступая одновременно в роли буферной памяти и обрабатываемого элемента. Информация в таких голограммах записывается путем изменения показателя преломления по всему объему носителя. Обычно для этого используются прозрачные материалы с ярко выраженными нелинейными оптическими характеристиками, например, кристаллы ниобата лития.

На рис. 64 показано примерное устройство оптической памяти с объемной голографической средой. Ввод информации осуществляется с помощью управляемого оптического транспаранта. Адресацией при записи-считывании управляет опорный луч. Считываемая информация фокусируется в плоскости многоэлементного матричного фотоприемника D.

В силу того, что емкость голографической памяти огромна, а время выборки мало, кардинально меняется весь подход к организации вычислительного процесса. Например, можно вернуться к идее широкого использования поиска по справочникам и таблицам функций, схем принятия решений, таблиц умножения, наконец. Разумеется, алгоритмический подход к обработке информации в оптическом компьютере сохранится, но его основой будет язык более высокого уровня организации, ориентированный на параллельную обработку сложных структур данных.

Речь пойдет об интерфейсах. Может показаться, что оптическая вычислительная среда, обладая уникальной способностью одновременно обрабатывать большие массивы информации, как бы оторвана от хорошо разработанной и всенародно любимой электронной вычислительной инфраструктуры, в которой обработка и передача информации осуществляется последовательно. Однако на самом деле любой оптический компьютер должен быть обрамлен обычной электроникой.

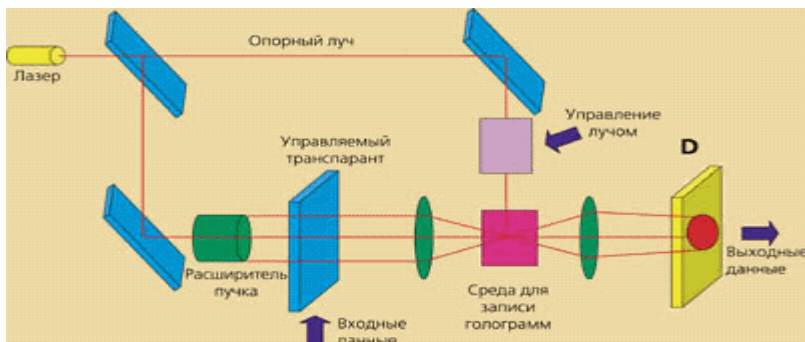


Рис. 64

Сегодня при вводе оптического изображения в ЭВМ вакуумные передающие телевизионные трубки полностью вытеснены приборами с зарядовой связью ПЗС (CCD - charge coupled device). Устройства на основе ПЗС воспринимают изображение параллельно по оптическому каналу, а передают последовательно по электронному. В ряде случаев последовательная перекачка содержимого ПЗС в оперативную память ЭВМ ограничивает быстродействие всего оптоэлектронного вычислительного комплекса. В настоящее время имеются российские и зарубежные разработки оптоэлектронных интерфейсов, в которых матричный фотоприемник конструктивно совмещен с элементами RAM в оперативной памяти ЭВМ. Отсюда - один шаг до использования оптического канала доступа к электронным чипам через третье измерение. Такие работы сегодня ведутся в зарубежных лабораториях. Оказывается, даже в стандартной персоналке длина соединений между вентилями микросхем, расположенных на разных печатных платах, порой измеряется метрами. Использование оптического канала для организации "междучипового" обмена позволяет на порядок сократить задержку распространения сигнала.

От ЭВМ к оптической системе информация передается с помощью управляемых транспарантов, или пространственных модуляторов света ПМС (SLM - spatial light modulator). Однако жидкокристаллические панели, чаще всего используемые для этих целей, обладают существенным недостатком - сравнительно низким быстродействием. Существуют также оптические транспаран-

ты с электронным управлением на цилиндрических магнитных доменах и термопластиках.

К числу перспективных направлений относится разработка ПМС с так называемыми умными ячейками (SP, smart pixels). Они могут модулировать амплитуду падающей световой волны как с использованием сигналов электронного интерфейса, так и с помощью второй световой волны. По сути - это прозрачная интегральная схема, выполненная по оптоэлектронной технологии и использующая параллельный оптический вход и выход информации с помощью третьего измерения. Интересно, что для первых экспериментов с устройствами типа SLM-SP обычные чипы кремниевой памяти покрывали тонким слоем жидкого кристалла. Электрическое поле вблизи заряженных ячеек памяти ориентирует молекулы жидкого кристалла, обнаруживая электрооптический эффект в отраженном свете. Впоследствии для этой цели стали разрабатывать специальные чипы, а также применять электро-магнитооптические жидкости и использовать эффект Фарадея. Такие транспаранты оказались более быстродействующими.

Еще один класс интерфейсных устройств образуют многоэлементные лазеры. Последнее достижение в этой области - матрицы полупроводниковых лазеров размером 1000x1000 светоизлучающих элементов. Излучение генерируется перпендикулярно плоскости чипа. В таких оптоэлектронных БИС используется технология изготовления лазеров с вертикальным расположением резонаторов (VCSEL, vertical cavity surface-emitting lasers). Недостаток подобных устройств состоит в том, что в каждый момент времени свет может излучаться только лазерами, расположенными либо в одной строке, либо в одном столбце матрицы.

По-видимому, в дальнейших разработках следует ожидать объединения технологий SLM-SP и VCSEL.

Правда жизни на стороне "аналоговой" природы: в некоторых наиболее эффективных своих применениях оптический процессор является аналоговой машиной. С другой стороны, способность оптической вычислительной среды образовывать параллельные связи между большим числом элементов и одновременно выполнять операции типа взвешивания и суммирования аналоговых сигналов является идеальной для построения нейросетей. Если в сис-

тему ввести транспарант, пропускание которого задано значениями весовых коэффициентов $W(i)$, получим простейшую модель нейрона - персептрон, показанный на рис. 65.

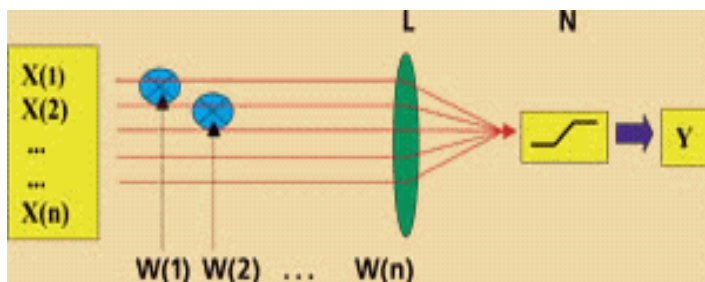


Рис. 65

Аналогично схему, представленную на рис. 65, можно рассматривать как слой нейронов. Для этого элементы фотоприемника должны иметь пороговую передаточную характеристику, а транспарант должен задавать таблицу весовых коэффициентов. Причем матрица преобразования совершенно не обязана быть квадратной. Более того, оптическими средствами можно создать связи между нейронами, организованными в двумерные массивы. Матрица связей при этом (формально) будет четырехмерной.

Для работы нейросети низкая точность вычислений, присущая аналоговым компьютерам, не имеет большого значения.

Объединяя рассмотренные оптические схемы в многослойные структуры и вводя в них обратные связи, можно создавать сложные быстродействующие нейросети. А на их основе - нейрокompьютеры широкого применения. Это направление считается одним из самых перспективных в области оптического компьютеростроения.

Успехи оптоэлектроники последнего десятилетия позволили достаточно хорошо отработать технологию производства всех основных компонентов оптического компьютера. Уже сейчас его можно было бы эффективно использовать в качестве сателлитного устройства к обычной персональной ЭВМ. Однако массового производства оптических компьютеров для широкого использования не наблюдается. Причина первая: долгое время работы в этой об-

ласти были закрытыми. Вторая причина проистекает из первой: никто не занимался маркетинговой проработкой оптического компьютера для народа.

Стоимость оптического компьютера довольно велика. Главные области его использования - военная техника, криминология, защита информации, банковская сфера, а также научные исследования. Основные работы за рубежом ведутся в США, Японии, Западной Европе. В США одним из ведущих координаторов работ является известное Агентство по передовым оборонным исследовательским проектам (DARPA, Defence Advanced Research Projects Agency). То самое Агентство, которое стояло у истоков Bitnet, прототипа Интернета. Исследования и разработки ведутся также в интересах ВВС, ВМС, НАСА, Организации СОИ и др. Решаемые задачи: машинное зрение, искусственный интеллект, распознавание и одновременное сопровождение сотен и тысяч целей, навигация, контроль земной поверхности, связь.

Из последних разработок в гражданском секторе отметим оптический процессор для проверки подлинности кредитных карт, паспортов и других важных документов. Дело в том, что жулики научились читать и подделывать обычные отражательные голограммы.

Сотрудники Коннектикутского университета и Оптического научного центра Аризонского университета предложили преобразовывать изображение отпечатков пальцев, лица, подписи владельца документа в специальный фазовый профиль, который одновременно записывается в прозрачной голографической среде на карте и в базе данных. Проверить подлинность карты можно лишь с помощью оптического процессора или коррелятора. Подделать фазовый профиль на прозрачной пластинке практически невозможно.

Уже в 1996 году фирмой IBM была разработана голографическая память. Для ввода картинок в память использовался управляемый транспарант размером 1024x1024 пикселей, способный работать с частотой 1 тыс. изображений в секунду. Для вывода информации применялась камера ПЗС размером 2048x2048 пикселей.

Фирмой OptiComp (США) разработан оптоэлектронный 32-битный RISC-процессор, способный обеспечить независимые соединения между 8192 оптическими каналами. Его производительность составляет около 10^{12} двоичных операций в секунду. Работая в режиме поиска текста, процессор может просматривать большие базы данных со скоростью 80 тыс. страниц (на каждой странице по 5 тыс. знаков) в секунду.

В совместной разработке Колорадского университета и исследовательского института JILA (США) оптический нейрокомпьютер используется для распознавания человеческой речи. Фурье-спектры отрезков человеческой речи преобразовывались в визуальную картину, которая далее обрабатывалась оптическими методами. Система работала гораздо быстрее и надежнее, чем специально обученная нейросеть на электронных чипах.

Несмотря на определенные сложности в разработке и изготовлении оптических компьютерных систем, идея его создания не умерла. Более того, сейчас она снова стала актуальной. Дело в том, что, увеличивая и увеличивая быстродействие процессора, производители с той же неизбежностью вынуждены работать со все более и более мелкими объектами. Сейчас отметка находится на рубеже 0,06 мкм, но это, разумеется, не предел. Нанотехнология, которая вначале тоже рассматривалась как некая полуфантастическая технология, постепенно превратилась в обычную задачу для современных технологов. Вот тут и выяснилось, что в нанотехнологии оптические подходы очень даже хороши.

Это можно объяснить следующим. С быстродействием у оптических компонентов все отлично. Если тактовая частота современных компьютерных систем только сейчас подошла к 3-4 ГГц, то оптические переключатели преодолели этот рубеж еще лет двадцать назад. Эффективность работы оптических переключателей можно видеть на примере волоконно-оптических линий связи. Здесь оптическая технология окончательно победила электрическую.

Наибольшее распространение получили переключатели на основе LiNbO_3 . Ниобат лития - уникальный кристалл. У него отличные электрооптические характеристики, он обладает замечательными акустооптическими свойствами и вдобавок ко всему еще

и нелинейными характеристиками. Поэтому в системах обработки оптической информации устройства на основе ниобата лития встречаются чаще всего.

В области создания элементов памяти для оптических систем тоже достигнуты значительные успехи, более того, в области оптической памяти в последние годы получены особенно интересные результаты. Активизация этих работ была стимулирована развитием носителей информации, а именно лазерных компакт-дисков и систем магнитооптической записи. Поскольку CD и магнитооптические диски уже стали коммерческими продуктами, то крупные фирмы имеют возможность наряду с финансированием чисто прикладных задач поддерживать и другие исследования в области оптической памяти.

Магнитооптическая память уже сейчас является вполне доступным продуктом. При этом стандартная плотность записи коммерческого диска составляет около 1 бит на 3 мкм² (что соответствует 128 Мбайт для 3,5-дюймового диска).

Исследования в этой области продвинулись сейчас очень далеко. В Information Storage Technology Group of MESA+ Institute (университет Твенте, г. Энсхеде, Нидерланды) успешно проведена работа по детектированию бита информации в области 5 Ангстрем. Счет идет просто на отдельные атомы. Полученные результаты действительно впечатляют: если эти разработки дойдут до промышленного производства, то 3,5-дюймовые диски емкостью в десятки гигабайт станут вполне доступны.

Есть еще одно направление оптической памяти - голографическая память. И здесь надо снова вспомнить о LiNbO₃. Мы уже говорили о нем, когда речь шла о системах оптической обработки информации. Разумеется, оставить такой выдающийся кристалл без внимания было нельзя. В последние годы на нем научились получать лазерную генерацию (легировав кристалл эрбием), а сейчас создают целые системы голографической памяти. Число голограмм на одном кристалле достигло десяти тысяч, и это не предел.

В оптических системах проблема интеграции является наиболее важной и в конечном итоге решающей. Когда все компоненты разработаны и испытаны, возникает очередная проблема - собрать все воедино. Для оптического компьютера эта проблема зна-

чительно сложнее, чем для обычного. Электрические связи реализовать гораздо проще. Можно, конечно, все элементы объединить с помощью оптического волокна (и такие технологии тоже развиваются), но в таких системах существуют довольно значительные потери информации. Поэтому не прекращаются попытки сделать все на одной "платформе".

Конечно, первое, на что ориентируются разработчики, так это кремний. Технология изготовления, обработки и производства разработаны до последних мелочей, существуют огромные производственные мощности. Нужна лишь база для производства. Поэтому любые новые оптические разработки на основе кремния принимаются обычно на ура. Но у кремния есть один хронический недостаток: он плохо совместим с активными оптическими элементами. Оптические усилители и переключатели очень плохо уживаются с кремниевой технологией.

Следующее направление - создание базы на основе альтернативных полупроводниковых структур. Здесь есть свои очень важные плюсы: можно формировать структуры с совершенно разными свойствами, встраивать их в оптические интегральные схемы (ОИС), нет проблем с источником света, поскольку ОИС сами могут генерировать лазерное излучение. Более того, не так давно произошел прорыв в области полупроводниковых лазеров, и они сейчас безусловные лидеры продаж. Казалось бы, вот готовая платформа для оптических компьютеров. Но есть две проблемы.

Первая - технологическая. Она связана со сложностью переориентации разработчиков систем на кремниевой основе на совершенно новое направление. Вторая - экономическая. Главной причиной убедительной победы кремниевой технологии была дешевизна. Кремния на земном шаре много и он достаточно дешев для использования в массовом производстве. Для новых структур требуются материалы гораздо более редкие и, как следствие, более дорогие, например галлий. Да и технология гораздо дороже.

Возникает законный вопрос: так будет ли в ближайшем будущем оптический компьютер, или нет. У оптических компьютеров есть свой "троянский конь", и имя ему - Internet. То, что в ближайшие годы именно Internet будет определять положение на компьютерном рынке, сейчас понимают почти все. При передаче

информации главной проблемой является трафик. Число пользователей растет, объем информации тоже. И здесь оптика уже полностью и окончательно одержала верх над "классической" электроникой. И сейчас не только вся информация перегоняется по оптическому кабелю, но и системы сопровождения тоже стали оптическими: электроника просто не тянет таких скоростей передачи и обработки информации. Пропускная же способность оптических систем растет, и возможности роста колоссальны.

Один из основных способов увеличения пропускной способности - системы спектрального уплотнения. И не зря было упомянуто, что на структурах из ниобата лития делают не только переключатели, лазерные усилители и голограммы. На них создают системы спектрального уплотнения. Почему бы не предположить, что вслед за чисто оптическими системами передачи Internet-информации могут прийти и чисто оптические системы обработки информации.

8.6.10. Нейронная архитектура

Нейрокомпьютер - это вычислительная система с архитектурой MSIMD, в которой реализованы два принципиальных технических решения: упрощен до уровня нейрона процессорный элемент однородной структуры и резко усложнены связи между элементами; программирование вычислительной структуры перенесено на изменение весовых связей между процессорными элементами.

Общее определение нейрокомпьютера может быть представлено в следующем виде. Нейрокомпьютер - это вычислительная система с архитектурой аппаратного и программного обеспечения, адекватной выполнению алгоритмов, представленных в нейросетевом логическом базисе

Каждый нейрон получает сигналы от соседних нейронов по специальным нервным волокнам. Эти сигналы могут быть возбуждающими или тормозящими. Их сумма составляет электрический потенциал внутри тела нейрона. Когда потенциал превышает некоторый порог, нейрон переходит в возбужденное состояние и посылает сигнал по выходному нервному волокну. Отдельные искус-

ственные нейроны соединяются друг с другом различными методами. Это позволяет создавать разнообразные нейронные сети с различной архитектурой, правилами обучения и возможностями.

Термин “искусственные нейронные сети” у многих ассоциируется с фантазиями об андроидах и бунте роботов, о машинах, заменяющих и имитирующих человека. Это впечатление усиливают многие разработчики нейросистем, рассуждая о том, как в недалеком будущем, роботы начнут осваивать различные виды деятельности, просто наблюдая за человеком. Если переключиться на уровень повседневной работы, то нейронные сети это всего-навсего сети, состоящие из связанных между собой простых элементов формальных нейронов. Большая часть работ по нейроинформатике посвящена переносу различных алгоритмов решения задач на такие сети.

В основу концепции положена идея о том, что нейроны можно моделировать довольно простыми автоматами, а вся сложность мозга, гибкость его функционирования и другие важнейшие качества определяются связями между нейронами. Каждая связь представляется как совсем простой элемент, служащий для передачи сигнала. Коротко эту мысль можно выразить так: “структура связей все, свойства элементов ничто”.

Совокупность идей и научно-техническое направление, определяемое описанным представлением о мозге, называется коннекционизмом (connection связь). С реальным мозгом все это соотносится примерно так же, как карикатура или шарж со своим прототипом. Важно не буквальное соответствие оригиналу, а продуктивность технической идеи.

С коннекционизмом тесно связан следующий блок идей:

- однородность системы (элементы одинаковы и чрезвычайно просты, все определяется структурой связей);
- надежные системы из ненадежных элементов и “аналоговый ренессанс” использование простых аналоговых элементов;
- “голографические” системы при разрушении случайно выбранной части система сохраняет свои свойства.

Предполагается, что широкие возможности систем связей компенсируют бедность выбора элементов, их ненадежность и возможные разрушения части связей.

Для описания алгоритмов и устройств в нейроинформатике выработана специальная “схемотехника”, в которой элементарные устройства (сумматоры, синапсы, нейроны и т.п.) объединяются в сети, предназначенные для решения задач. Для многих начинающих кажется неожиданным, что ни в аппаратной реализации нейронных сетей, ни в профессиональном программном обеспечении эти элементы вовсе не обязательно реализуются как отдельные части или блоки. Используемая в нейроинформатике идеальная схемотехника представляет собой особый язык описания нейронных сетей и их обучения. При программной и аппаратной реализации выполненные на этом языке описания переводятся на более подходящие языки другого уровня.

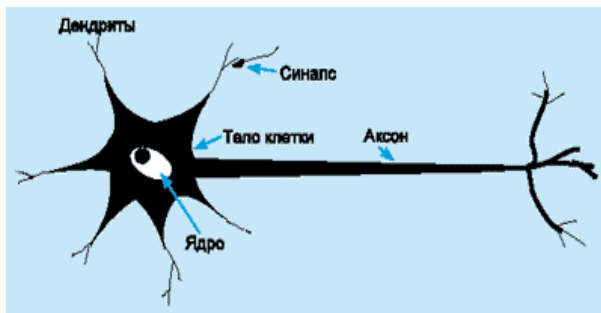


Рис. 66

Нейрон (нервная клетка) является особой биологической клеткой, которая обрабатывает информацию (рис. 66). Она состоит из тела клетки (cell body), или сомы (soma), и двух типов внешних древоподобных ветвей: аксона (axon) и дендритов (dendrites). Тело клетки включает ядро (nucleus), которое содержит информацию о наследственных свойствах, и плазму, обладающую молекулярными средствами для производства необходимых нейрону материалов. Нейрон получает сигналы (импульсы) от других нейронов через дендриты (приемники) и передает сигналы, сгенерированные телом клетки, вдоль аксона (передатчик), который в конце разветвляется на волокна (strands). На окончаниях этих волокон находятся синапсы (synapses).

Синапс является элементарной структурой и функциональным узлом между двумя нейронами (волокно аксона одного нейрона и дендрит другого). Когда импульс достигает синаптического окончания, высвобождаются определенные химические вещества, называемые нейротрансмиттерами. Нейротрансмиттеры диффундируют через синаптическую щель, возбуждая или затормаживая, в зависимости от типа синапса, способность нейрона-приемника генерировать электрические импульсы. Результативность синапса может настраиваться проходящими через него сигналами, так что синапсы могут обучаться в зависимости от активности процессов, в которых они участвуют. Эта зависимость от предыстории действует как память, которая, возможно, ответственна за память человека.

Кора головного мозга человека является протяженной, образованной нейронами поверхностью толщиной от 2 до 3 мм с площадью около 2200 см^2 , что вдвое превышает площадь поверхности стандартной клавиатуры. Кора головного мозга содержит около 1011 нейронов, что приблизительно равно числу звезд Млечного пути. Каждый нейрон связан с 103 - 104 другими нейронами. В целом мозг человека содержит приблизительно от 1014 до 1015 взаимосвязей.

Нейроны взаимодействуют посредством короткой серии импульсов, как правило, продолжительностью несколько мсек. Сообщение передается посредством частотно-импульсной модуляции. Частота может изменяться от нескольких единиц до сотен герц, что в миллион раз медленнее, чем самые быстродействующие переключательные электронные схемы. Тем не менее, сложные решения по восприятию информации, как, например, распознавание лица, человек принимает за несколько сотен мс. Эти решения контролируются сетью нейронов, которые имеют скорость выполнения операций всего несколько мс. Это означает, что вычисления требуют не более 100 последовательных стадий. Другими словами, для таких сложных задач мозг "запускает" параллельные программы, содержащие около 100 шагов. Это известно как правило ста шагов. Рассуждая аналогичным образом, можно обнаружить, что количество информации, посылаемое от одного нейрона другому, должно быть очень маленьким (несколько бит). От-

сюда следует, что основная информация не передается непосредственно, а захватывается и распределяется в связях между нейронами.

История создания искусственных нейронов уходит своими корнями в 1943 год, когда шотландец МакКаллок и англичанин Питтс создали теорию формальных нейросетей, а через пятнадцать лет Розенблатт изобрел искусственный нейрон (персептрон), который, впоследствии, и лег в основу нейрокомпьютера.

Искусственный нейрон имитирует в первом приближении свойства биологического нейрона. На вход искусственного нейрона поступает некоторое множество сигналов, каждый из которых является выходом другого нейрона. Каждый вход умножается на соответствующий вес, аналогичный синаптической силе, и все произведения суммируются, определяя уровень активации нейрона.

На рис. 67 представлена модель реализующая эту идею. Хотя сетевые парадигмы весьма разнообразны, в основе почти всех их лежит эта конфигурация. Здесь множество входных сигналов, обозначенных $x_1, x_2, x_3 \dots x_n$, поступает на искусственный нейрон. Эти входные сигналы, в совокупности обозначаемые вектором \mathbf{X} , соответствуют сигналам, приходящим в синапсы биологического нейрона. Каждый сигнал умножается на соответствующий вес $w_1, w_2, w_3 \dots w_n$, и поступает на суммирующий блок (Σ) (адаптивный сумматор). Каждый вес соответствует "силе" одной биологической синаптической связи (множество весов в совокупности обозначается вектором \mathbf{W}).

Суммирующий блок, соответствующий телу биологического элемента, складывает взвешенные входы алгебраически, создавая на выходе сигнал \mathbf{XW} . Он преобразуется активационной функцией \mathbf{F} и дает выходной нейронный сигнал \mathbf{OUT} .

Активационная функция может быть обычной линейной функцией:

$$\mathbf{OUT} = \mathbf{K}(\mathbf{XW}),$$

где \mathbf{K} - постоянная, пороговой функцией,

$\mathbf{OUT} = 1$, если $\mathbf{XW} > \mathbf{T}$,

$\mathbf{OUT} = 0$ в остальных случаях,

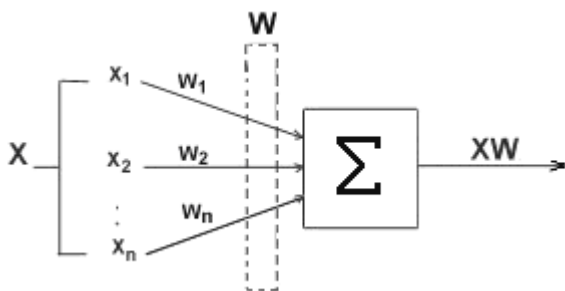


Рис. 67.

где T - некоторая постоянная пороговая величина, или же функцией более точно моделирующей нелинейную передаточную характеристику биологического нейрона и представляющей нейронной сети большие возможности.

На рис. 68 блок обозначенный F , принимает сигнал XW и выдает сигнал OUT . Если блок F сужает диапазон изменения величины XW так, что при любых значениях XW значения OUT принадлежат некоторому конечному интервалу, то F называется сжимающей функцией.

В качестве сжимающей функции часто используется логистическая или сигмоидальная (S-образная) функция, показанная на рис. 69. Эта функция математически выражается как

$$F(x)=1/(1+e^{-x}).$$

Таким образом, $OUT=1/(1+e^{-XW})$.

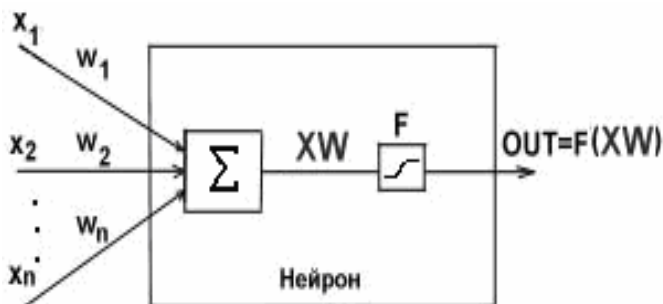


Рис. 68

По аналогии с электронными системами активационную функцию можно считать нелинейной усилительной характеристикой искусственного нейрона. Коэффициент усиления вычисляется как отношение приращения величины OUT к вызвавшему его небольшому приращению величины XW . Он выражается наклоном кривой при определенном уровне возбуждения и изменяется от малых значений при больших отрицательных возбуждениях (кривая почти горизонтальна) до максимального значения при нулевом возбуждении и снова уменьшается, когда возбуждение становится большим положительным.

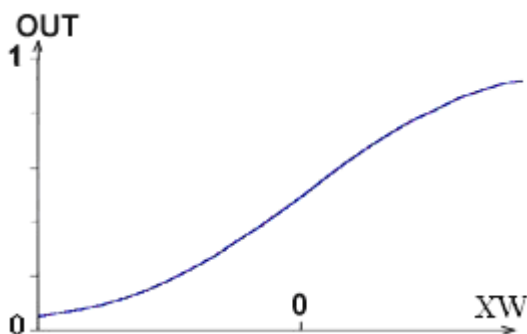


Рис. 69.

Гроссберг (1973) обнаружил, что подобная нелинейная характеристика решает поставленную им дилемму шумового насыщения. Слабые сигналы нуждаются в большом сетевом усилении, чтобы дать пригодный к использованию выходной сигнал. Однако усилительные каскады с большими коэффициентами усиления могут привести к насыщению выхода шумами усилителей (случайными флуктуациями), которые присутствуют в любой физически реализованной сети. Сильные входные сигналы в свою очередь также будут приводить к насыщению усилительных каскадов, исключая возможность полезного использования выхода. Центральная область логистической функции, имеющая большой коэффициент усиления, решает проблему обработки слабых сигналов, в то время как в области с падающим усилением на положительном и

отрицательном концах подходят для больших возбуждений. Таким образом, нейрон функционирует с большим усилением в широком диапазоне уровня входного сигнала.

Рассмотренная простая модель искусственного нейрона игнорирует многие свойства своего биологического двойника. Например, она не принимает во внимание задержки во времени, которые воздействуют на динамику системы. Входные сигналы сразу же порождают выходной сигнал. И что более важно, она не учитывает воздействий функции частотной модуляции или синхронизирующей функции биологического нейрона, которые ряд исследователей считают решающими. Несмотря на эти ограничения, сети, построенные из этих нейронов, обнаруживают свойства, сильно напоминающие биологическую систему. Только время и исследования смогут ответить на вопрос, являются ли подобные совпадения случайными или следствием того, что в модели верно схвачены важнейшие черты биологического нейрона

Каждый нейрон получает сигналы от соседних нейронов по специальным нервным волокнам. Эти сигналы могут быть возбуждающими или тормозящими. Их сумма составляет электрический потенциал внутри тела нейрона. Когда потенциал превышает некоторый порог, нейрон переходит в возбужденное состояние и посылает сигнал по выходному нервному волокну. Отдельные искусственные нейроны соединяются друг с другом различными методами. Это позволяет создавать разнообразные нейронные сети с различной архитектурой, правилами обучения и возможностями.

Искусственная Нейронная Сеть (ИНС) может рассматриваться как направленный граф со взвешенными связями, в котором искусственные нейроны являются узлами. По архитектуре связей ИНС могут быть сгруппированы в два класса: сети прямого распространения, в которых графы не имеют петель, и рекуррентные сети, или сети с обратными связями.

В наиболее распространенном семействе сетей первого класса, называемых многослойным перцептроном, нейроны расположены слоями и имеют однонаправленные связи между слоями. На рисунке представлены типовые сети каждого класса. Сети прямого распространения являются статическими в том смысле, что на заданный вход они вырабатывают одну совокупность вы-

ходных значений, не зависящих от предыдущего состояния сети. Рекуррентные сети являются динамическими, так как в силу обратных связей в них модифицируются входы нейронов, что приводит к изменению состояния сети.

Здесь каждый нейрон передает свой выходной сигнал остальным нейронам, включая самого себя. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети. Все входные сигналы подаются всем нейронам. Элементы слоистых и полносвязных сетей могут выбираться по-разному. Существует, впрочем, стандартный выбор: нейрон с адаптивным неоднородным линейным сумматором на входе. Для полносвязной сети входной сумматор нейрона фактически распадается на два: первый вычисляет линейную функцию от входных сигналов сети, второй линейную функцию от выходных сигналов других нейронов, полученных на предыдущем шаге. Функция активации нейронов (характеристическая функция) это нелинейный преобразователь выходного сигнала сумматора. Если функция одна для всех нейронов сети, то сеть называют однородной (гомогенной). Если же характеристическая функция зависит еще от одного или нескольких параметров, значения которых меняются от нейрона к нейрону, то сеть называют неоднородной (гетерогенной).

Составлять сеть из нейронов стандартного вида не обязательно. Слоистая или полносвязная архитектуры не налагают существенных ограничений на участвующие в них элементы. Единственное жесткое требование, предъявляемое архитектурой к элементам сети, это соответствие размерности вектора входных сигналов элемента (она определяется архитектурой) числу его входов. Если полносвязная сеть функционирует до получения ответа заданное число тактов k , то ее можно представить как частный случай k -слойной сети, все слои которой одинаковы и каждый из них соответствует такту функционирования полносвязной сети.

Существенное различие между полносвязной и слоистой сетями становится очевидным, когда число тактов функционирования заранее не ограничено и слоистая сеть так работать не может. Доказаны теоремы о полноте: для любой непрерывной функции нескольких переменных можно построить нейронную сеть,

которая вычисляет эту функцию с любой заданной точностью. Так что нейронные сети в каком-то смысле могут все.

Способность к обучению является фундаментальным свойством мозга. В контексте ИНС процесс обучения может рассматриваться как настройка архитектуры сети и весов связей для эффективного выполнения специальной задачи. Обычно нейронная сеть должна настроить веса связей по имеющейся обучающей выборке. Функционирование сети улучшается по мере итеративной настройки весовых коэффициентов. Свойство сети обучаться на примерах делает их более привлекательными по сравнению с системами, которые следуют определенной системе правил функционирования, сформулированной экспертами.

Для конструирования процесса обучения, прежде всего, необходимо иметь модель внешней среды, в которой функционирует нейронная сеть - знать доступную для сети информацию. Эта модель определяет парадигму обучения. Во-вторых, необходимо понять, как модифицировать весовые параметры сети - какие правила обучения управляют процессом настройки. Алгоритм обучения означает процедуру, в которой используются правила обучения для настройки весов.

Существуют три парадигмы обучения: "с учителем", "без учителя" (самообучение) и смешанная. В первом случае нейронная сеть располагает правильными ответами (выходами сети) на каждый входной пример. Веса настраиваются так, чтобы сеть производила ответы как можно более близкие к известным правильным ответам. Усиленный вариант обучения с учителем предполагает, что известна только критическая оценка правильности выхода нейронной сети, но не сами правильные значения выхода. Обучение без учителя не требует знания правильных ответов на каждый пример обучающей выборки. В этом случае раскрывается внутренняя структура данных или корреляции между образцами в системе данных, что позволяет распределить образцы по категориям. При смешанном обучении часть весов определяется посредством обучения с учителем, в то время как остальная получается с помощью самообучения.

Теория обучения рассматривает три фундаментальных свойства, связанных с обучением по примерам: емкость, слож-

ность образцов и вычислительная сложность. Под емкостью понимается, сколько образцов может запомнить сеть, и какие функции и границы принятия решений могут быть на ней сформированы. Сложность образцов определяет число обучающих примеров, необходимых для достижения способности сети к обобщению. Слишком малое число примеров может вызвать "переобученность" сети, когда она хорошо функционирует на примерах обучающей выборки, но плохо - на тестовых примерах, подчиненных тому же статистическому распределению.

Известны 4 основных типа правил обучения: коррекция по ошибке, машина Больцмана, правило Хебба и обучение методом соревнования.

- правило коррекции по ошибке. При обучении с учителем для каждого входного примера задан желаемый выход d . Реальный выход сети y может не совпадать с желаемым. Принцип коррекции по ошибке при обучении состоит в использовании сигнала $(d-y)$ для модификации весов, обеспечивающей постепенное уменьшение ошибки. Обучение имеет место только в случае, когда перцептрон ошибается. Известны различные модификации этого алгоритма обучения.

- обучение Больцмана представляет собой стохастическое правило обучения, которое следует из информационных теоретических и термодинамических принципов. Целью обучения Больцмана является такая настройка весовых коэффициентов, при которой состояния видимых нейронов удовлетворяют желаемому распределению вероятностей. Обучение Больцмана может рассматриваться как специальный случай коррекции по ошибке, в котором под ошибкой понимается расхождение корреляций состояний в двух режимах .

- правило Хебба: если нейроны с обеих сторон синапса активизируются одновременно и регулярно, то сила синаптической связи возрастает. Важной особенностью этого правила является то, что изменение синаптического веса зависит только от активности нейронов, которые связаны данным синапсом. Это существенно упрощает цепи обучения в реализации VLSI.

- обучение методом соревнования. В отличие от обучения Хебба, в котором множество выходных нейронов могут возбуж-

даться одновременно, при соревновательном обучении выходные нейроны соревнуются между собой за активизацию. Это явление известно как правило "победитель берет все". Подобное обучение имеет место в биологических нейронных сетях. Обучение посредством соревнования позволяет кластеризовать входные данные: подобные примеры группируются сетью в соответствии с корреляциями и представляются одним элементом. При обучении модифицируются только веса "победившего" нейрона. Эффект этого правила достигается за счет такого изменения сохраненного в сети образца (вектора весов связей победившего нейрона), при котором он становится чуть ближе к входному примеру.

Одной из особенностей нейросетевых методов обработки информации является высокая параллельность вычислений и, следовательно, целесообразность использования специальных средств аппаратной поддержки. В значительной мере успех в решении рассмотренных задач обусловлен использованием оригинальных ускорительных плат. Такие платы работают параллельно с процессором обыкновенного ПК и несут на себе основную вычислительную нагрузку, превращая основной процессор компьютера в устройство управления и обслуживания мощных вычислительных средств, расположенных на ускорительной плате.

Например в НТЦ "Модуль" разработаны многопроцессорные ускорительные платы МЦ5.001 и МЦ5.002. Первая из них имеет в своем составе 4 микропроцессора TMS320C40 с тактовой частотой 50 МГц и пиковой производительностью 275 MIPS. Каждый процессор имеет свою локальную статическую память объемом 1 Мбайт. К 2 процессорам дополнительно подключены 2 блока динамической памяти объемом 16 Мбайт каждый. К одному из процессоров подключена также статическая память объемом 1 Мбайт, используемая для обмена данными с ПК. Процессоры соединены друг с другом специальными высокоскоростными каналами с пропускной способностью 20 Мбайт/с каждый. Нарращивание и комплексирование плат осуществляется на материнской плате ПК с помощью шины ISA.

Ускорительная плата МЦ5.002 содержит 6 процессоров TMS320C40 и выполнена в конструктиве VME, что позволяет ис-

пользовать ее в бортовых системах, расположенных на летательном аппарате.

Нейропроцессор обычно состоит из двух основных блоков: скалярного, выполняющего роль универсального вычислительного устройства, и векторного, ориентированного на выполнение векторно-матричных операций. Скалярное устройство обеспечивает интерфейсы с памятью и коммуникационными портами, позволяющими объединять процессоры в вычислительные сети различной конфигурации. Основное назначение скалярного устройства - подготовка данных для векторной части процессора. Для этого существует несколько режимов адресации, интерфейс с памятью, наборы арифметических и логических операций, возможность работы с регистровыми парами.

Центральным звеном нейропроцессора является целочисленное векторное устройство, обладающее возможностями обработки данных различной разрядности. Оно оперирует n -разрядными словами. Таким образом, процессор рассчитан на высокопроизводительную обработку больших массивов целочисленных данных. К примеру отечественный нейропроцессор, разработанный в НТЦ "Модуль". Скалярное устройство обеспечивает интерфейсы с памятью и 2 коммуникационными портами. Скалярное устройство имеет адресных регистров и такое же количество регистров общего назначения разрядностью 32 бита каждый.

Центральным звеном нейропроцессора является целочисленное векторное устройство, обладающее возможностями обработки данных различной разрядности. Оно оперирует 64-разрядными словами, которые могут быть разбиты на целочисленные составляющие практически произвольной разрядности в пределах от 1 до 64 бит. На каждую инструкцию векторного процессора затрачивается от 1 до 32 тактов. При этом одновременно обрабатывается до 32 64-разрядных слов. Для организации непрерывной подачи данных в операционное устройство (ОУ) векторного процессора используются внутренние блоки памяти, называемые векторными регистрами. Они выполняют роль буфера операндов, буфера для хранения матрицы весов, очереди результатов. При выполнении команды в операционном устройстве операнды по очереди извлекаются из внутреннего буфера и подаются на

один из входов ОУ. Внутри ОУ производятся вычисления, а их результат заносится в буфер результатов. Векторные инструкции, хотя и занимают несколько тактов процессорного времени, могут выполняться параллельно с инструкциями скалярного процессора. Таким образом, процессор рассчитан на высокопроизводительную обработку больших массивов целочисленных данных.

Нейропроцессор выполнен по технологии 0,5 мкм. Его тактовая частота 33 МГц. На специальных векторно-матричных операциях он дает увеличение производительности в десятки раз по сравнению с процессором TMS320C40. Благодаря наличию коммуникационных портов с интерфейсом, идентичным портам TMS320C40, нейропроцессор может быть интегрирован в гетерогенную многопроцессорную систему.

Для нейропроцессора разработан полный пакет системного программного обеспечения, включая символьный отладчик, и ряд прикладных библиотек, в частности библиотеку векторно-матричных вычислений.

Специфика рассматриваемых вычислительных средств и решаемых задач обуславливает новые требования к технике программирования. Программисту приходится оперировать другими категориями, по-другому строить логику программы, решать задачи, которые не могли возникнуть при традиционном программировании. Перед ним стоит задача - максимально эффективно использовать ресурсы вычислительной системы, правильно распределить нагрузку между процессорами, задействовать их специфические возможности.

Здесь на первый план выходят методы параллельной обработки данных. Причем слова "параллельная обработка" можно понимать как обработку на параллельно работающих процессорах, так и одновременную обработку нескольких элементов данных на одном процессоре. Современный процессор позволяет выполнять несколько инструкций за один такт, что заставляет программиста продумывать как способы организации самих вычислений, так и способы подготовки данных, для того чтобы параллельно выполняемые процессы не блокировали друг друга.

Трудности, возникающие при программировании многопроцессорных систем, хорошо известны: синхронизация парал-

лельных процессов, механизмы обмена данными, проблемы "критических участков", когда несколько процессов задействуют одни и те же ресурсы. Еще одной важной особенностью современных процессоров является высокая разрядность операндов, например 64 бита, что позволяет размещать в них по несколько малоразрядных элементов данных и обрабатывать их параллельно. Примером эффективного использования отмеченной особенности современных процессоров является технология MMX, где 64-разрядный регистр разбивается на 8 независимых байтов или на 4 16-битных слова, которые обрабатываются параллельно. Независимость элементов состоит в том, что при смещениях или вычитании не происходит заимствования битов у соседних элементов.

По сравнению с обычными компьютерами, нейрокомпьютеры обладают рядом преимуществ. Во первых — высокое быстродействие, связанное с тем, что алгоритмы нейроинформатики обладают высокой степенью параллельности. Во вторых — нейросистемы делаются очень устойчивыми к помехам и разрушениям. В третьих — устойчивые и надежные нейросистемы могут создаваться из ненадежных элементов, имеющих значительный разброс параметров.

Несмотря на перечисленные выше преимущества, эти устройства имеют ряд недостатков:

1. Они создаются специально для решения конкретных задач, связанных с нелинейной логикой и теорией самоорганизации. Решение таких задач на обычных компьютерах возможно только численными методами.

2. В силу своей уникальности эти устройства достаточно дорогостоящи.

Иллюстрацией преимуществ нейросистем по сравнению с другими их типами может быть диаграмма, представленная на рис.70.

Большинство современных нейросистем представляют собой просто персональный компьютер или рабочую станцию, в состав которых входит дополнительная нейроплата. К их числу относятся, например, компьютеры серии FMR фирмы Fujitsu. Такие системы имеют бесспорное право на существование, поскольку их возможностей вполне достаточно для разработки новых алгорит-

мов и решения большого числа прикладных задач методами нейроматематики.

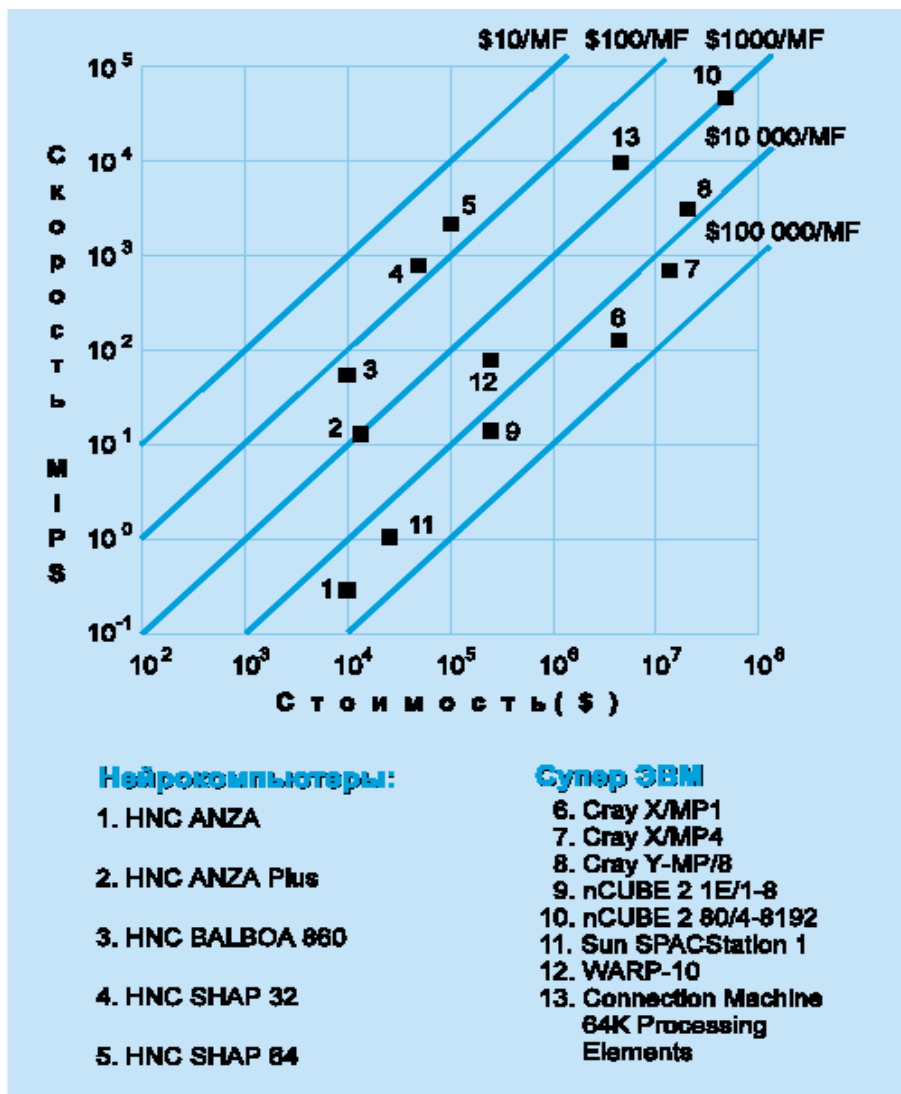


Рис. 70

Однако наибольший интерес представляют специализированные нейрокомпьютеры, непосредственно реализующие принципы НС. Типичными представителями таких систем являются

компьютеры семейства Mark фирмы TRW (первая реализация перцептрона, разработанная Розенблатом, называлась Mark I).

Модель Mark III фирмы TRW представляют собой рабочую станцию, содержащую до 15 процессоров семейства Motorola 68000 с математическими сопроцессорами. Все процессоры объединены шиной VME. Архитектура системы, поддерживающая до 65 000 виртуальных процессорных элементов с более чем 1 млн. настраиваемых соединений, позволяет обрабатывать до 450 тыс. межсоединений/с.

Mark IV - это однопроцессорный суперкомпьютер с конвейерной архитектурой. Он поддерживает до 236 тыс. виртуальных процессорных элементов, что позволяет обрабатывать до 5 млн. межсоединений/с.

Компьютеры семейства Mark имеют общую программную оболочку ANSE (Artificial Neural System Environment), обеспечивающую программную совместимость моделей.

Помимо указанных моделей фирмы TRW предлагает также пакет Mark II - программный эмулятор НС.

Другой интересной моделью является нейрокомпьютер NETSIM, созданный фирмой Texas Instruments на базе разработок Кембриджского университета. Его топология представляет собой трехмерную решетку стандартных вычислительных узлов на базе процессоров 80188. Компьютер NETSIM используется для моделирования таких моделей НС, как сеть Хопфилда - Кохонена и НС с обратным распространением. Его производительность достигает 450 млн. межсоединений/с.

Фирма Computer Recognition Systems (CRS) продает серию нейрокомпьютеров WIZARD/CRS 1000, предназначенных для обработки видеоизображений. Размер входной изображения 512 x 512 пиксел. Модель CRS 1000 уже нашла применение в промышленных системах автоматического контроля.

8.6.11. Масштабируемая архитектура

Ярким примером масштабируемой архитектуры является система Cray T3E, которая использует микропроцессоры DEC 21164 (DEC Alpha EV5), RISK-процессоры с пиковой производи-

тельностью 600 Мфлоп и 21164А для машин Cray T3E-900 и Cray T3E-1200. Каждый



процессорный элемент (ПЭ) Cray T3E имеет свою собственную DRAM-память объёмом от 64 Мбайт до 2 Гбайт. В отличие от системы CRAY T3D, в которой исполняемая задача запрашивает фиксированное количество процессоров на все время выполнения, в CRAY T3E неиспользуемые процессоры могут использоваться другими задачами. Модели T3E, T3E-900, T3E-1200, T3E-1350.

Каждый узел в системе содержит один процессорный элемент (ПЭ), включающий процессор, память и средство коммутации, которое осуществляет связь между ПЭ. Система конфигурируется до 2048 процессоров. Пиковая производительность составляет 2,4 Тфлоп.

Разделяемая, высокопроизводительная, глобально адресуемая подсистема памяти делает возможным обращение к локальной памяти каждого ПЭ в Cray T3E. Процессорные элементы в системе Cray T3E связаны в трехмерный тор двунаправленной высокоскоростной сетью с малым временем задержки, которая в шесть раз превосходит по скорости аналогичную сеть в Cray T3D. Также добавлена адаптивная маршрутизация, при которой возможен обход участков с высокой эффективностью передачи.

Cray T3E выполняют операции ввода/вывода через многочисленные порты на один или более каналы GigaRing. Каналы ввода/вывода интегрированы в 3-х мерную межзловую сеть и пропорциональны размеру системы. При этом при добавлении ПЭ пропускная способность каналов ввода/вывода увеличивается и масштабируемые приложения могут выполняться на системах с большим числом процессоров также эффективно, как на системах с меньшим числом процессоров.

Для Cray T3E была создана масштабируемая версия операционной системы ОС UNICOS — ОС UNICOS/mk. Операционная система UNICOS/mk разделена на программы-серверы, распределенные среди процессоров Cray T3E. Это позволяет управлять набором ресурсов системы как единым целым. Локальные серверы обрабатывают запросы ОС, специфичные для каждого ПЭ. Глобальные серверы обеспечивают общесистемные возможности такие, как управление процессами и файловые операции.

В добавлении к пользовательским ПЭ, которые выполняют приложения и команды, системы Cray T3E включает специальные системные ПЭ, которые выполняют глобальные сервера UNICOS/mk. Так как глобальные сервера расположены на системных ПЭ и не дублируются по всей системе, UNICOS/mk эффективно масштабируема, полно функциональна и обслуживает от десятков до тысячи ПЭ с минимальным перегрузкой.

UNICOS/mk обеспечивает следующие программные функции:

- Распределение серверов управления файлами. Функции файлового сервера распределены, используя локальные файловые программы-сервера, для обеспечения максимальной производительности и эффективности.

- ПЭ может генерировать не только последовательную, но и параллельную передачу данных, используя некоторые или даже все ПЭ данной программы.

- Множество глобальных файловых серверов: Система управления файлами распределена на множество системных ПЭ, которые позволяют полностью использовать параллельные дисковые каналы, поддерживаемые на Cray T3E.

Cray T3E-1200 в два раза превышает производительность систем Cray T3E при уменьшенной вдвое стоимости за Мфлоп. Конфигурации в воздушно-жидкостном охлаждении имеют от 6-и процессоров, а в жидкостном — от 32 процессоров. Каждый процессор имеет производительность в 1,2 Тфлоп, для всей системы пиковая производительность меняется от 7,2 Гфлоп до 2,5 Тфлоп. Масштабируется до тысяч процессоров. Серия выпущена в 1997 году.

Система предназначена для наиболее важных научных и технических задач в аэрокосмической, автомобильной, финансовой, химико-фармацевтической, нефтяной и т.д. промышленности, также в широких областях прикладных исследований, включая химию, гидродинамику, предсказание погоды и сейсмические процессы.

Для поддержки масштабируемости используется оперативная система UNICOS/mk — масштабируемая версия UNICOSR. ТЗЕ-1200 поддерживает как явное распараллеливание распределённой памяти посредством CF90 и C/C++ с передачей сообщений (MPI, MPI-2 и PVM) и передачу данных, так и неявное распараллеливание посредством возможностей HPF и Cray CRAFT.

ТЗЕ выполняет операции ввода/вывода через многочисленные порты на один и более каналов посредством интерфейса GigaRing. Каждый канал сдвоенного кольца ввода/вывода, содержащий в двух кольцах данные, которые перемещаются в противоположных направлениях, передают данные ввода/вывода с высокой пропускной способностью и повышенной надёжностью. Все каналы ввода/вывода доступны и управляемы всеми процессорными элементами. В ТЗЕ каждый интерфейс GigaRing имеет максимальную пропускную способность в 500 Мбайт/с.

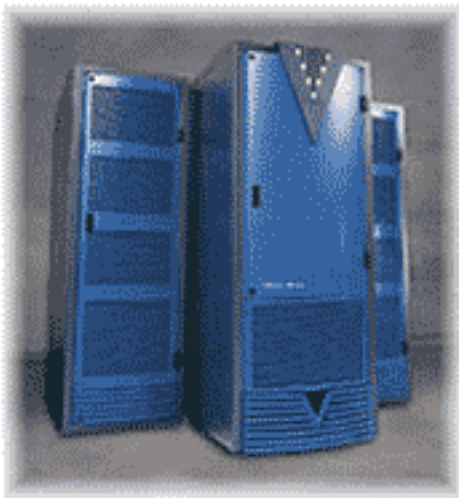
В дополнение к высокой производительности и пропускной способности процессорных элементов и высокой масштабируемости, Cray ТЗЕ-1200 имеет две уникальные особенности: STREAMS и E-Регистры. STREAMS доводят до максимума пропускную способность локальной памяти, позволяя микропроцессору запускать при полной скорости для ссылки для вектороподобных данных. E-Регистры предоставляют операции gather/scatter (соединение/вразброс) для ссылок на локальную и удалённую память, и используют полную пропускную способность внутреннего соединения для удалённого чтения и записи отдельного слова.

Оценка производительности системы производилась при решении плотной линейной системы уравнений порядка 148800 на машине ТЗЕ-1200 с 1200 процессорами. Была достигнута скорость в 1,127 Тфлоп/с, что составляет 63% эффективности.

9. Многопроцессорные системы

9.1. Общие требования, предъявляемые к МПС

Отношение стоимость / производительность. Появление любого нового направления в вычислительной технике определяется требованиями



компьютерного рынка. Поэтому у разработчиков компьютеров нет одной единственной цели. Большая универсальная вычислительная машина (мейнфрейм) или суперкомпьютер стоят дорого. Для достижения поставленных целей при проектировании высокопроизводительных конструкций приходится игнорировать стоимостные характеристики.

Суперкомпьютеры фирмы Cray Research и высокопроизводительные мейнфреймы компании IBM относятся именно к этой категории компьютеров. Другим крайним примером может служить низкостоимостная конструкция, где производительность принесена в жертву для достижения низкой стоимости. К этому направлению относятся персональные компьютеры различных клонов IBM PC. Между этими двумя крайними направлениями находятся конструкции, основанные на отношении стоимость/ производительность, в которых разработчики находят баланс между стоимостными параметрами и производительностью. Типичными примерами такого рода компьютеров являются миникомпьютеры и рабочие станции.

Для сравнения различных компьютеров между собой обычно используются стандартные методики измерения производительности. Эти методики позволяют разработчикам и пользователям использовать полученные в результате испытаний количественные показатели для оценки тех или иных технических реше-

ний, и в конце концов именно производительность и стоимость дают пользователю рациональную основу для решения вопроса, какой компьютер выбрать.

Надежность и отказоустойчивость. Важнейшей характеристикой вычислительных систем является надежность. Повышение надежности основано на принципе предотвращения неисправностей путем снижения интенсивности отказов и сбоев за счет применения электронных схем и компонентов с высокой и сверхвысокой степенью интеграции, снижения уровня помех, облегченных режимов работы схем, обеспечение тепловых режимов их работы, а также за счет совершенствования методов сборки аппаратуры.

Отказоустойчивость - это такое свойство вычислительной системы, которое обеспечивает ей, как логической машине, возможность продолжения действий, заданных программой, после возникновения неисправностей. Введение отказоустойчивости требует избыточного аппаратного и программного обеспечения. Направления, связанные с предотвращением неисправностей и с отказоустойчивостью, - основные в проблеме надежности. Концепции параллельности и отказоустойчивости вычислительных систем естественным образом связаны между собой, поскольку в обоих случаях требуются дополнительные функциональные компоненты. Поэтому, собственно, на параллельных вычислительных системах достигается как наиболее высокая производительность, так и, во многих случаях, очень высокая надежность. Имеющиеся ресурсы избыточности в параллельных системах могут гибко использоваться как для повышения производительности, так и для повышения надежности. Структура многопроцессорных и многомашинных систем приспособлена к автоматической реконфигурации и обеспечивает возможность продолжения работы системы после возникновения неисправностей.

Следует помнить, что понятие надежности включает не только аппаратные средства, но и программное обеспечение. Главной целью повышения надежности систем является целостность хранимых в них данных.

Масштабируемость. Масштабируемость представляет собой возможность наращивания числа и мощности процессоров,

объемов оперативной и внешней памяти и других ресурсов вычислительной системы. Масштабируемость должна обеспечиваться архитектурой и конструкцией компьютера, а также соответствующими средствами программного обеспечения.

Добавление каждого нового процессора в действительно масштабируемой системе должно давать прогнозируемое увеличение производительности и пропускной способности при приемлемых затратах. Одной из основных задач при построении масштабируемых систем является минимизация стоимости расширения компьютера и упрощение планирования. В идеале добавление процессоров к системе должно приводить к линейному росту ее производительности. Однако это не всегда так. Потери производительности могут возникать, например, при недостаточной пропускной способности шин из-за возрастания трафика между процессорами и основной памятью, а также между памятью и устройствами ввода/вывода. В действительности реальное увеличение производительности трудно оценить заранее, поскольку оно в значительной степени зависит от динамики поведения прикладных задач.

Возможность масштабирования системы определяется не только архитектурой аппаратных средств, но зависит от заложенных свойств программного обеспечения. Масштабируемость программного обеспечения затрагивает все его уровни от простых механизмов передачи сообщений до работы с такими сложными объектами как мониторы транзакций и вся среда прикладной системы. В частности, программное обеспечение должно минимизировать трафик межпроцессорного обмена, который может препятствовать линейному росту производительности системы. Аппаратные средства (процессоры, шины и устройства ввода/вывода) являются только частью масштабируемой архитектуры, на которой программное обеспечение может обеспечить предсказуемый рост производительности. Важно понимать, что простой переход, например, на более мощный процессор может привести к перегрузке других компонентов системы. Это означает, что действительно масштабируемая система должна быть сбалансирована по всем параметрам.



Совместимость и мобильность программного обеспечения. Концепция программной совместимости впервые в широких масштабах была применена разработчиками системы ИВМ/360. Основная задача при проектировании всего ряда моделей этой системы заключалась в создании такой архитектуры, которая была бы одинаковой с точки зрения пользователя для всех моделей системы независимо от цены и производительности каждой из них. Огромные преимущества такого подхода, позволяющего сохранять существующий задел программного обеспечения при переходе на новые (как правило, более производительные) модели были быстро оценены как производителями компьютеров, так и пользователями и начиная с этого времени практически все фирмы-поставщики компьютерного оборудования взяли на вооружение эти принципы, поставляя серии совместимых компьютеров. Следует заметить однако, что со временем даже самая передовая архитектура неизбежно устаревает и возникает потребность внесения радикальных изменений архитектуру и способы организации вычислительных систем.

В настоящее время одним из наиболее важных факторов, определяющих современные тенденции в развитии информационных технологий, является ориентация компаний-поставщиков компьютерного оборудования на рынок прикладных программных средств. Это объясняется прежде всего тем, что для конечного пользователя в конце концов важно программное обеспечение, позволяющее решить его задачи, а не выбор той или иной аппарат-

ной платформы. Переход от однородных сетей программно совместимых компьютеров к построению неоднородных сетей, включающих компьютеры разных фирм-производителей, в корне изменил и точку зрения на саму сеть: из сравнительно простого средства обмена информацией она превратилась в средство интеграции отдельных ресурсов - мощную распределенную вычислительную систему, каждый элемент которой (сервер или рабочая станция) лучше всего соответствует требованиям конкретной прикладной задачи.

Этот переход выдвинул ряд новых требований. Прежде всего такая вычислительная среда должна позволять гибко менять количество и состав аппаратных средств и программного обеспечения в соответствии с меняющимися требованиями решаемых задач. Во-вторых, она должна обеспечивать возможность запуска одних и тех же программных систем на различных аппаратных платформах, т.е. обеспечивать мобильность программного обеспечения. В третьих, эта среда должна гарантировать возможность применения одних и тех же человеко-машинных интерфейсов на всех компьютерах, входящих в неоднородную сеть. В условиях жесткой конкуренции производителей аппаратных платформ и программного обеспечения сформировалась концепция открытых систем, представляющая собой совокупность стандартов на различные компоненты вычислительной среды, предназначенных для обеспечения мобильности программных средств в рамках неоднородной, распределенной вычислительной системы.

Одним из вариантов моделей открытой среды является модель OSE (Open System Environment), предложенная комитетом IEEE POSIX. На основе этой модели национальный институт стандартов и технологии США выпустил документ "Application Portability Profile (APP). The U.S. Government's Open System Environment Profile OSE/1 Version 2.0", который определяет рекомендуемые для федеральных учреждений США спецификации в области информационных технологий, обеспечивающие мобильность системного и прикладного программного обеспечения. Все ведущие производители компьютеров и программного обеспечения в США в настоящее время придерживаются требований этого документа.

9.2. Классификация систем параллельной обработки данных

На протяжении всей истории развития вычислительной техники делались попытки найти какую-то общую классификацию, под которую попадали бы все возможные направления развития компьютерных архитектур. Ни одна из таких классификаций не могла охватить все разнообразие разрабатываемых архитектурных решений и не выдерживала испытания временем. Тем не менее в научный оборот попали и широко используются ряд терминов, которые полезно знать не только разработчикам, но и пользователям компьютеров.

Любая вычислительная система (будь то супер-ЭВМ или персональный компьютер) достигает своей наивысшей производительности благодаря использованию высокоскоростных элементов и параллельному выполнению большого числа операций. Именно возможность параллельной работы различных устройств системы (работы с перекрытием) является основой ускорения основных операций.

Параллельные ЭВМ часто подразделяются по классификации Флинна на машины типа SIMD и MIMD. Как и любая другая, приведенная выше классификация несовершенна: существуют машины прямо в нее не попадающие, имеются также важные признаки, которые в этой классификации не учтены. В частности, к машинам типа SIMD часто относят векторные процессоры, хотя их высокая производительность зависит от другой формы параллелизма - конвейерной организации машины. Многопроцессорные векторные системы, типа Стру Y-MP, состоят из нескольких векторных процессоров и поэтому могут быть названы MSIMD (Multiple SIMD).

Классификация Флинна не делает различия по другим важным для вычислительных моделей характеристикам, например, по уровню "зернистости" параллельных вычислений и методам синхронизации.

Можно выделить четыре основных типа архитектуры систем параллельной обработки:

- **конвейерная и векторная обработка.** Основу конвейерной обработки составляет раздельное выполнение некоторой операции в несколько этапов (за несколько ступеней) с передачей данных одного этапа следующему. Производительность при этом возрастает благодаря тому, что одновременно на различных ступенях конвейера выполняются несколько операций.

Конвейеризация эффективна только тогда, когда загрузка конвейера близка к полной, а скорость подачи новых операндов соответствует максимальной производительности конвейера. Если происходит задержка, то параллельно будет выполняться меньше операций и суммарная производительность снизится. Векторные операции обеспечивают идеальную возможность полной загрузки вычислительного конвейера.

При выполнении векторной команды одна и та же операция применяется ко всем элементам вектора (или чаще всего к соответствующим элементам пары векторов). Для настройки конвейера на выполнение конкретной операции может потребоваться некоторое установочное время, однако затем операнды могут поступать в конвейер с максимальной скоростью, допускаемой возможностями памяти. При этом не возникает пауз ни в связи с выборкой новой команды, ни в связи с определением ветви вычислений при условном переходе. Таким образом, главный принцип вычислений на векторной машине состоит в выполнении некоторой элементарной операции или комбинации из нескольких элементарных операций, которые должны повторно применяться к некоторому блоку данных. Таким операциям в исходной программе соответствуют небольшие компактные циклы.

- **системы типа SIMD.** Машины типа SIMD состоят из большого числа идентичных процессорных элементов, имеющих собственную память. Все процессорные элементы в такой машине выполняют одну и ту же программу. Очевидно, что такая машина, составленная из большого числа процессоров, может обеспечить очень высокую производительность только на тех задачах, при решении которых все процессоры могут делать одну и ту же работу. Модель вычислений для машины SIMD очень похожа на модель вычислений для векторного процессора: одиночная операция выполняется над большим блоком данных.

В отличие от ограниченного конвейерного функционирования векторного процессора, матричный процессор (синоним для большинства SIMD-машин) может быть значительно более гибким. Обработываемые элементы таких процессоров - это универсальные программируемые ЭВМ, так что задача, решаемая параллельно, может быть достаточно сложной и содержать ветвления. Обычное проявление этой вычислительной модели в исходной программе примерно такое же, как и в случае векторных операций: циклы на элементах массива, в которых значения, вырабатываемые на одной итерации цикла, не используются на другой итерации цикла.

Модели вычислений на векторных и матричных ЭВМ настолько схожи, что эти ЭВМ часто обсуждаются как эквивалентные.

- **системы типа MIMD.** Термин "мультипроцессор" покрывает большинство машин типа MIMD и (подобно тому, как термин "матричный процессор" применяется к машинам типа SIMD) часто используется в качестве синонима для машин типа MIMD. В мультипроцессорной системе каждый процессорный элемент (ПЭ) выполняет свою программу достаточно независимо от других процессорных элементов.

Процессорные элементы, конечно, должны как-то связываться друг с другом, что делает необходимым более подробную классификацию машин типа MIMD. В мультипроцессорах с общей памятью (сильносвязанных мультипроцессорах) имеется память данных и команд, доступная всем ПЭ. С общей памятью ПЭ связываются с помощью общей шины или сети обмена. В противоположность этому варианту в слабосвязанных многопроцессорных системах (машинах с локальной памятью) вся память делится между процессорными элементами и каждый блок памяти доступен только связанному с ним процессору. Сеть обмена связывает процессорные элементы друг с другом.

Базовой моделью вычислений на MIMD-мультипроцессоре является совокупность независимых процессов, эпизодически обращающихся к разделяемым данным. Существует большое количество вариантов этой модели. На одном конце спектра - модель распределенных вычислений, в которой программа делится на до-

вольно большое число параллельных задач, состоящих из множества подпрограмм. На другом конце спектра - модель потоковых вычислений, в которых каждая операция в программе может рассматриваться как отдельный процесс. Такая операция ждет своих входных данных (операндов), которые должны быть переданы ей другими процессами. По их получении операция выполняется, и полученное значение передается тем процессам, которые в нем нуждаются. В потоковых моделях вычислений с большим и средним уровнем гранулярности, процессы содержат большое число операций и выполняются в потоковой манере.

- **многопроцессорные системы с SIMD-процессорами.** Многие современные супер-ЭВМ представляют собой многопроцессорные системы, в которых в качестве процессоров используются векторные процессоры или процессоры типа SIMD. Такие системы относятся к машинам класса MSIMD.

Языки программирования и соответствующие компиляторы для машин типа MSIMD обычно обеспечивают языковые конструкции, которые позволяют программисту описывать "крупнозернистый" параллелизм. В пределах каждой задачи компилятор автоматически векторизует подходящие циклы. Машины типа MSIMD, как можно себе представить, дают возможность использовать лучший из этих двух принципов декомпозиции: векторные операции ("мелкозернистый" параллелизм) для тех частей программы, которые подходят для этого, и гибкие возможности MIMD-архитектуры для других частей программы.

Многопроцессорные системы за годы развития вычислительной техники претерпели ряд этапов своего развития. Исторически первой стала осваиваться технология SIMD. Однако в настоящее время наметился устойчивый интерес к архитектурам MIMD. Этот интерес главным образом определяется двумя факторами:

- архитектура MIMD дает большую гибкость: при наличии адекватной поддержки со стороны аппаратных средств и программного обеспечения MIMD может работать как однопользовательская система, обеспечивая высокопроизводительную обработку данных для одной прикладной задачи, как многопрограммная

машина, выполняющая множество задач параллельно, и как некоторая комбинация этих возможностей.

- архитектура MIMD может использовать все преимущества современной микропроцессорной технологии на основе строгого учета соотношения стоимость/производительность. В действительности практически все современные многопроцессорные системы строятся на тех же микропроцессорах, которые можно найти в персональных компьютерах, рабочих станциях и небольших однопроцессорных серверах.

Одной из отличительных особенностей многопроцессорной вычислительной системы является сеть обмена, с помощью которой процессоры соединяются друг с другом или с памятью. Модель обмена настолько важна для многопроцессорной системы, что многие характеристики производительности и другие оценки выражаются отношением времени обработки к времени обмена, соответствующим решаемым задачам. Существуют две основные модели межпроцессорного обмена: одна основана на передаче сообщений, другая - на использовании общей памяти.

В многопроцессорной системе с общей памятью один процессор осуществляет запись в конкретную ячейку, а другой процессор производит считывание из этой ячейки памяти. Чтобы обеспечить согласованность данных и синхронизацию процессов, обмен часто реализуется по принципу взаимно исключающего доступа к общей памяти методом "почтового ящика".

В архитектурах с локальной памятью непосредственное разделение памяти невозможно. Вместо этого процессоры получают доступ к совместно используемым данным посредством передачи сообщений по сети обмена. Эффективность схемы коммуникаций зависит от протоколов обмена, основных сетей обмена и пропускной способности памяти и каналов обмена.

Часто, и притом необосновано, в машинах с общей памятью и векторных машинах затраты на обмен не учитываются, так как проблемы обмена в значительной степени скрыты от программиста. Однако накладные расходы на обмен в этих машинах имеются и определяются конфликтами шин, памяти и процессоров. Чем больше процессоров добавляется в систему, тем больше процессов соперничают при использовании одних и тех же данных и шины,

что приводит к состоянию насыщения. Модель системы с общей памятью очень удобна для программирования и иногда рассматривается как высокоуровневое средство оценки влияния обмена на работу системы, даже если основная система в действительности реализована с применением локальной памяти и принципа передачи сообщений.

В сетях с коммутацией каналов и в сетях с коммутацией пакетов по мере возрастания требований к обмену следует учитывать возможность перегрузки сети. Здесь межпроцессорный обмен связывает сетевые ресурсы: каналы, процессоры, буферы сообщений. Объем передаваемой информации может быть сокращен за счет тщательной функциональной декомпозиции задачи и тщательного диспетчирования выполняемых функций.

Таким образом, существующие MIMD-машины распадаются на два основных класса в зависимости от количества объединяемых процессоров, которое определяет и способ организации памяти и методику их межсоединений.

К первой группе относятся машины с общей (разделяемой) основной памятью, объединяющие до нескольких десятков (обычно менее 32) процессоров. Сравнительно небольшое количество процессоров в таких машинах позволяет иметь одну централизованную общую память и объединить процессоры и память с помощью одной шины. При наличии у процессоров кэш-памяти достаточного объема высокопроизводительная шина и общая память могут удовлетворить обращения к памяти, поступающие от нескольких процессоров. Поскольку имеется единственная память с одним и тем же временем доступа, эти машины иногда называются UMA (Uniform Memory Access). Такой способ организации со сравнительно небольшой разделяемой памятью в настоящее время является наиболее популярным. Структура подобной системы представлена на рис. 66.

Вторую группу машин составляют крупномасштабные системы с распределенной памятью. Для того чтобы поддерживать большое количество процессоров приходится распределять основную память между ними, в противном случае полосы пропускания памяти просто может не хватить для удовлетворения запросов, поступающих от очень большого числа процессоров. Естественно

при таком подходе также требуется реализовать связь процессоров между собой. На рис. 67 показана структура такой системы.

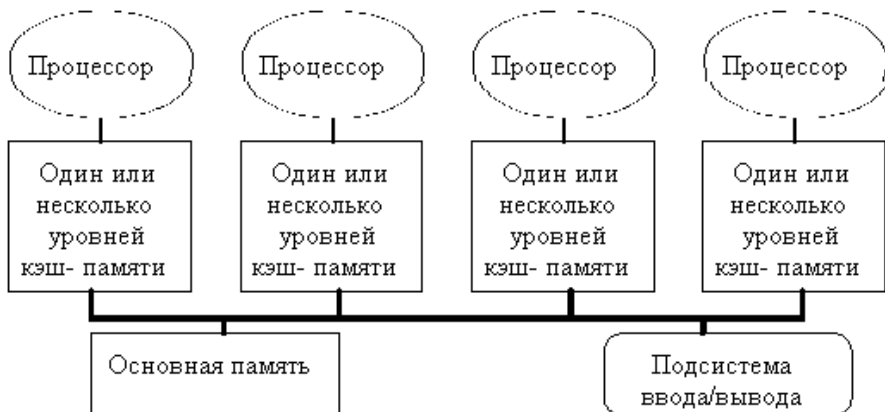


Рис. 66

С ростом числа процессоров просто невозможно обойти необходимость реализации модели распределенной памяти с высокоскоростной сетью для связи процессоров. С быстрым ростом производительности процессоров и связанным с этим ужесточением требования увеличения полосы пропускания памяти, масштаб систем (т.е. число процессоров в системе), для которых требуется организация распределенной памяти, уменьшается, также как и уменьшается число процессоров, которые удается поддерживать на одной разделяемой шине и общей памяти.

Распределение памяти между отдельными узлами системы имеет два главных преимущества. Во-первых, это эффективный с точки зрения стоимости способ увеличения полосы пропускания памяти, поскольку большинство обращений могут выполняться параллельно к локальной памяти в каждом узле. Во-вторых, это уменьшает задержку обращения (время доступа) к локальной памяти. Эти два преимущества еще больше сокращают количество процессоров, для которых архитектура с распределенной памятью имеет смысл.

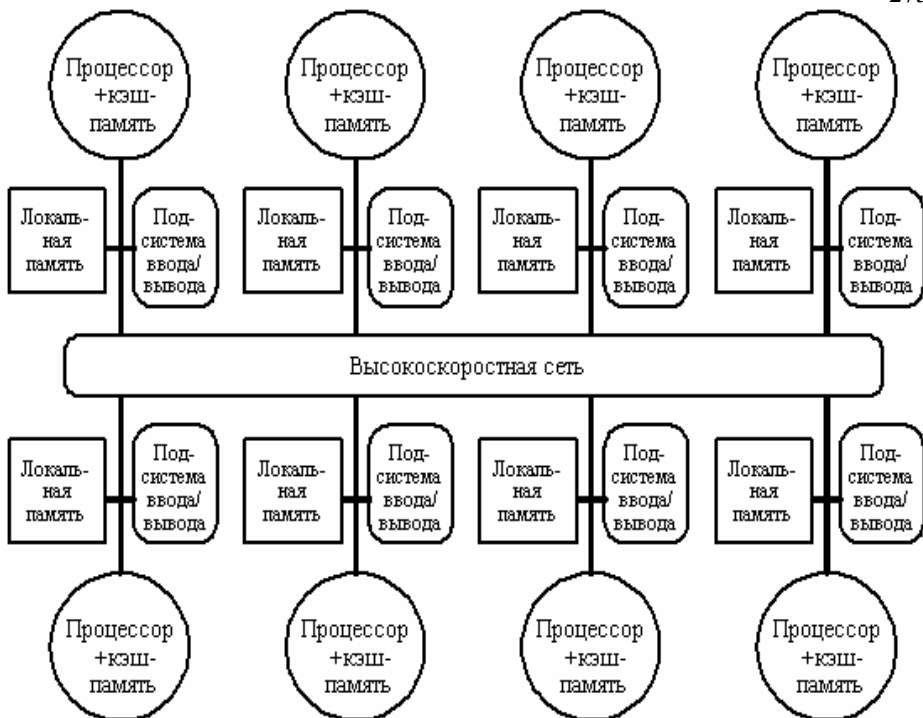


Рис. 67

Обычно устройства ввода/вывода, также как и память, распределяются по узлам и в действительности узлы могут состоять из небольшого числа (2-8) процессоров, соединенных между собой другим способом. Хотя такая кластеризация нескольких процессоров с памятью и сетевой интерфейс могут быть достаточно полезными с точки зрения эффективности в стоимостном выражении, это не очень существенно для понимания того, как такая машина работает, поэтому мы пока остановимся на системах с одним процессором на узел. Основная разница в архитектуре, которую следует выделить в машинах с распределенной памятью заключается в том, как осуществляется связь и какова логическая модель памяти.

9.3. Модели связи и архитектуры памяти

Как уже было отмечено, любая крупномасштабная многопроцессорная система должна использовать множество устройств памяти, которые физически распределяются вместе с процессорами. Имеется две альтернативных организации адресации этих устройств памяти и связанных с этим два альтернативных метода для передачи данных между процессорами. Физически отдельные устройства памяти могут адресоваться как логически единое адресное пространство, что означает, что любой процессор может выполнять обращения к любым ячейкам памяти, предполагая, что он имеет соответствующие права доступа. Такие машины называются машинами с распределенной разделяемой (общей) памятью (DSM - distributed shared memory), масштабируемые архитектуры с разделяемой памятью, а иногда NUMA's - Non-Uniform Memory Access, поскольку время доступа зависит от расположения ячейки в памяти.

В альтернативном случае, адресное пространство состоит из отдельных адресных пространств, которые логически не связаны и доступ к которым не может быть осуществлен аппаратно другим процессором. В таком примере каждый модуль процессор-память представляет собой отдельный компьютер, поэтому такие системы называются многомашинными (multicomputers).

С каждой из этих организаций адресного пространства связан свой механизм обмена. Для машины с единым адресным пространством это адресное пространство может быть использовано для обмена данными посредством операций загрузки и записи. Поэтому эти машины и получили название машин с разделяемой (общей) памятью. Для машин с множеством адресных пространств обмен данными должен использовать другой механизм: передачу сообщений между процессорами; поэтому эти машины часто называют машинами с передачей сообщений.

Каждый из этих механизмов обмена имеет свои преимущества. Для обмена в общей памяти это включает:

- совместимость с хорошо понятными используемыми как в однопроцессорных, так и маломасштабных многопроцессорных

системах, механизмами, которые используют для обмена общей памятью.

- простота программирования, когда модели обмена между процессорами сложные или динамически меняются во время выполнения. Подобные преимущества упрощают конструирование компилятора.

- более низкая задержка обмена и лучшее использование полосы пропускания при обмене малыми порциями данных.

- возможность использования аппаратно управляемого кэширования для снижения частоты удаленного обмена, допускающая кэширование всех данных как разделяемых, так и неразделяемых.

Основные преимущества обмена с помощью передачи сообщений являются:

- аппаратура может быть более простой, особенно по сравнению с моделью разделяемой памяти, которая поддерживает масштабируемую когерентность кэш-памяти.

- модели обмена понятны, принуждают программистов (или компиляторы) уделять внимание обмену, который обычно имеет высокую, связанную с ним стоимость.

Конечно, требуемая модель обмена может быть настроена над аппаратной моделью, которая использует любой из этих механизмов. Поддержка передачи сообщений над разделяемой памятью, естественно, намного проще, если предположить, что машины имеют адекватные полосы пропускания. Основные трудности возникают при работе с сообщениями, которые могут быть неправильно выровнены и сообщениями произвольной длины в системе памяти, которая обычно ориентирована на передачу выровненных блоков данных, организованных как блоки кэш-памяти. Эти трудности можно преодолеть либо с небольшими потерями производительности программным способом, либо существенно без потерь при использовании небольшой аппаратной поддержки.

Построение механизмов реализации разделяемой памяти над механизмом передачи сообщений намного сложнее. Без предполагаемой поддержки со стороны аппаратуры все обращения к разделяемой памяти потребуют привлечения операционной системы как для обеспечения преобразования адресов и защиты памяти,

так и для преобразования обращений к памяти в посылку и прием сообщений. Поскольку операции загрузки и записи обычно работают с небольшим объемом данных, то большие накладные расходы по поддержанию такого обмена делают невозможной чисто программную реализацию.

При оценке любого механизма обмена критичными являются три характеристики производительности:

1. *Полоса пропускания*: в идеале полоса пропускания механизма обмена будет ограничена полосами пропускания процессора, памяти и системы межсоединений, а не какими-либо аспектами механизма обмена. Связанные с механизмом обмена накладные расходы (например, длина межпроцессорной связи) прямо воздействуют на полосу пропускания.

2. *Задержка*: в идеале задержка должна быть настолько мала, насколько это возможно. Для ее определения критичны накладные расходы аппаратуры и программного обеспечения, связанные с инициированием и завершением обмена.

3. *Упрятывание задержки*: насколько хорошо механизм скрывает задержку путем перекрытия обмена с вычислениями или с другими обменами.

Каждый из этих параметров производительности воздействует на характеристики обмена. В частности, задержка и полоса пропускания могут меняться в зависимости от размера элемента данных. В общем случае, механизм, который одинаково хорошо работает как с небольшими, так и с большими объемами данных будет более гибким и эффективным.

Таким образом, отличия разных машин с распределенной памятью определяются моделью памяти и механизмом обмена. Исторически машины с распределенной памятью первоначально были построены с использованием механизма передачи сообщений, поскольку это было очевидно проще и многие разработчики и исследователи не верили, что единое адресное пространство можно построить и в машинах с распределенной памятью. С недавнего времени модели обмена с общей памятью действительно начали поддерживаться практически в каждой разработанной машине (характерным примером могут служить системы с симметричной мультипроцессорной обработкой).

Хотя машины с централизованной общей памятью, построенные на базе общей шины все еще доминируют в терминах размера компьютерного рынка, долговременные технические тенденции направлены на использование преимуществ распределенной памяти даже в машинах умеренного размера. Как мы увидим, возможно наиболее важным вопросом, который встает при создании машин с распределенной памятью, является вопрос о кэшировании и когерентности кэш-памяти.

9.4. Многопроцессорные системы с общей памятью

Требования, предъявляемые современными процессорами к полосе пропускания памяти можно существенно сократить путем применения больших многоуровневых кэшей. Тогда, если эти требования снижаются, то несколько процессоров смогут разделять доступ к одной и той же памяти. Начиная с 1980 года эта идея, подкрепленная широким распространением микропроцессоров, стимулировала многих разработчиков на создание небольших мультипроцессоров, в которых несколько процессоров разделяют одну физическую память, соединенную с ними с помощью разделяемой шины.

Из-за малого размера процессоров и заметного сокращения требуемой полосы пропускания шины, достигнутого за счет возможности реализации достаточно большой кэш-памяти, такие машины стали исключительно эффективными по стоимости. В первых разработках подобного рода машин удавалось разместить весь процессор и кэш на одной плате, которая затем вставлялась в заднюю панель, с помощью которой реализовывалась шинная архитектура. Современные конструкции позволяют разместить до четырех процессоров на одной плате. В такой машине 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. Когда кэшируется элемент частных данных, их значение переносится в кэш для сокращения среднего времени доступа, а также требуемой полосы пропускания.

Поскольку никакой другой процессор не использует эти данные, этот процесс идентичен процессу для однопроцессорной машины с кэш-памятью. Если кэшируются разделяемые данные, то разделяемое значение реплицируется и может 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.

Мультипроцессорная когерентность кэш-памяти. Проблема, о которой идет речь, возникает из-за того, что значение элемента данных в памяти, хранящееся в двух разных процессорах, доступно этим процессорам только через их индивидуальные кэши. На рис. 68 показан простой пример, иллюстрирующий эту проблему.

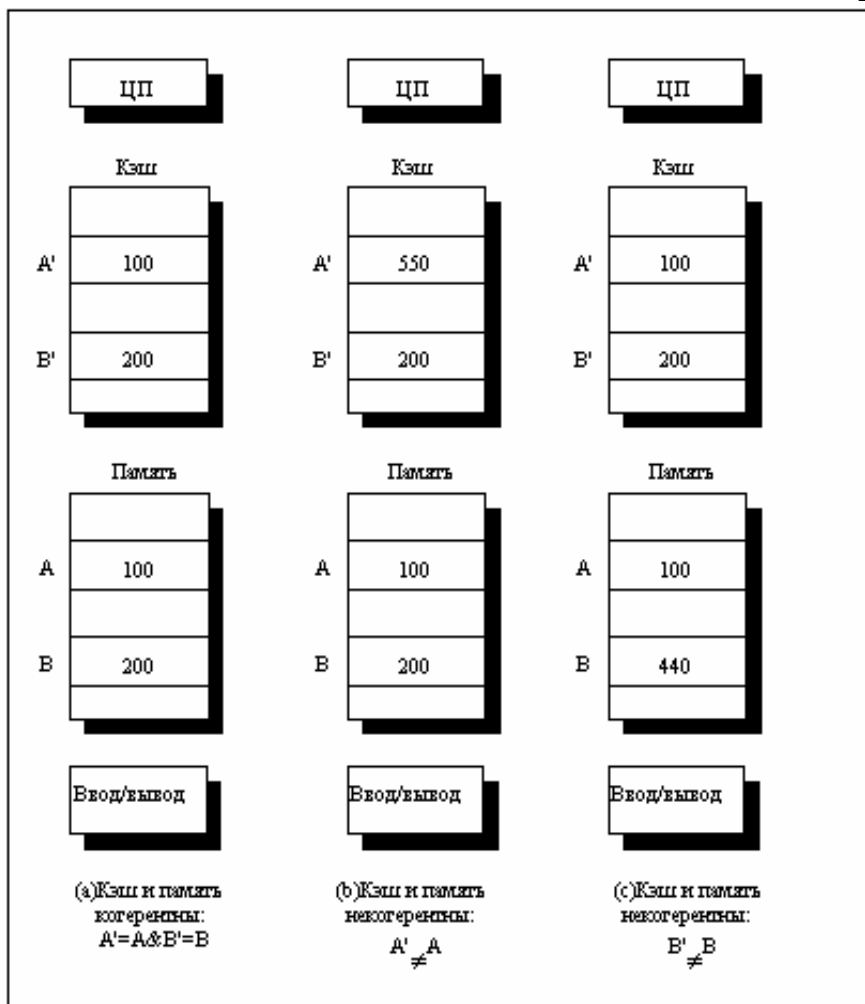
Проблема когерентности памяти для мультипроцессоров и устройств ввода/вывода имеет много аспектов. Обычно в малых мультипроцессорах используется аппаратный механизм, называемый протоколом, позволяющий решить эту проблему. Такие протоколы называются протоколами когерентности кэш-памяти.

Существуют два класса таких протоколов:

1. Протоколы на основе справочника (directory based). Информация о состоянии блока физической памяти содержится только в одном месте, называемом справочником (физически справочник может быть распределен по узлам системы). Этот подход будет рассмотрен в разд. 10.3.

2. Протоколы наблюдения (snooping). Каждый кэш, который содержит копию данных некоторого блока физической памяти, имеет также соответствующую копию служебной информации о его состоянии. Централизованная система записей отсутствует.

Обычно кэши расположены на общей (разделяемой) шине и контроллеры всех кэшей наблюдают за шиной (просматривают ее) для определения того, не содержат ли они копию соответствующего блока.



A' и B' - кэшированные копии элементов A и B в основной памяти

- Когерентное состояние кэша и основной памяти.
- Предполагается использование кэш-памяти с отложенным обратным копированием, 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.
- Подсистема ввода/вывода вводит в ячейку памяти B новое значение 440, а в кэш-памяти осталось старое значение B .

Рис. 68

В мультипроцессорных системах, использующих микропроцессоры с кэш-памятью, подсоединенные к централизованной общей памяти, протоколы наблюдения приобрели популярность, поскольку для опроса состояния кэшей они могут использовать заранее существующее физическое соединение - шину памяти.

Неформально, проблема когерентности памяти состоит в необходимости гарантировать, что любое считывание элемента данных возвращает последнее по времени записанное в него значение. Это определение не совсем корректно, поскольку невозможно требовать, чтобы операция считывания мгновенно видела значение, записанное в этот элемент данных некоторым другим процессором. Если, например, операция записи на одном процессоре предшествует операции чтения той же ячейки на другом процессоре в пределах очень короткого интервала времени, то невозможно гарантировать, что чтение вернет записанное значение данных, поскольку в этот момент времени записываемые данные могут даже не покинуть процессор.

Вопрос о том, когда точно записываемое значение должно быть доступно процессору, выполняющему чтение, определяется выбранной моделью согласованного (непротиворечивого) состояния памяти и связан с реализацией синхронизации параллельных вычислений. Поэтому с целью упрощения предположим, что мы требуем только, чтобы записанное операцией записи значение было доступно операции чтения, возникшей немного позже записи и что операции записи данного процессора всегда видны в порядке их выполнения.

С этим простым определением согласованного состояния памяти мы можем гарантировать когерентность путем обеспечения двух свойств:

1. Операция чтения ячейки памяти одним процессором, которая следует за операцией записи в ту же ячейку памяти другим процессором, получит записанное значение, если операции чтения и записи достаточно отделены друг от друга по времени.

2. Операции записи в одну и ту же ячейку памяти выполняются строго последовательно (иногда говорят, что они сериализованы): это означает, что две подряд идущие операции записи в од-

ну и ту же ячейку памяти будут наблюдаться другими процессорами именно в том порядке, в котором они появляются в программе процессора, выполняющего эти операции записи.

Первое свойство очевидно связано с определением когерентного (согласованного) состояния памяти: если бы процессор всегда бы считывал только старое значение данных, мы сказали бы, что память некогерентна.

Необходимость строго последовательного выполнения операций записи является более тонким, но также очень важным свойством. Представим себе, что строго последовательное выполнение операций записи не соблюдается. Тогда процессор P1 может записать данные в ячейку, а затем в эту ячейку выполнит запись процессор P2. Строго последовательное выполнение операций записи гарантирует два важных следствия для этой последовательности операций записи. Во-первых, оно гарантирует, что каждый процессор в машине в некоторый момент времени будет наблюдать запись, выполняемую процессором P2. Если последовательность операций записи не соблюдается, то может возникнуть ситуация, когда какой-нибудь процессор будет наблюдать сначала операцию записи процессора P2, а затем операцию записи процессора P1, и будет хранить это записанное P1 значение неограниченно долго.

Более тонкая проблема возникает с поддержанием разумной модели порядка выполнения программ и когерентности памяти для пользователя: представьте, что третий процессор постоянно читает ту же самую ячейку памяти, в которую записывают процессоры P1 и P2; он должен наблюдать сначала значение, записанное P1, а затем значение, записанное P2. Возможно, он никогда не сможет увидеть значения, записанного P1, поскольку запись от P2 возникла раньше чтения. Если он даже видит значение, записанное P1, он должен видеть значение, записанное P2, при последующем чтении.

Подобным образом любой другой процессор, который может наблюдать за значениями, записываемыми как P1, так и P2, должен наблюдать идентичное поведение. Простейший способ добиться таких свойств заключается в строгом соблюдении порядка операций записи, чтобы все записи в одну и ту же ячейку могли

наблюдаться в том же самом порядке. Это свойство называется последовательным выполнением (сериализацией) операций записи (write serialization). Вопрос о том, когда процессор должен увидеть значение, записанное другим процессором достаточно сложен и имеет заметное воздействие на производительность, особенно в больших машинах.

Альтернативные протоколы. Имеются две методики поддержания описанной выше когерентности. Один из методов заключается в том, чтобы гарантировать, что процессор должен получить исключительные права доступа к элементу данных перед выполнением записи в этот элемент данных. Этот тип протоколов называется протоколом записи с аннулированием (write invalidate protocol), поскольку при выполнении записи он аннулирует другие копии. Это наиболее часто используемый протокол как в схемах на основе справочников, так и в схемах наблюдения. Исключительное право доступа гарантирует, что во время выполнения записи не существует никаких других копий элемента данных, в которые можно писать или из которых можно читать: все другие кэшированные копии элемента данных аннулированы.

Чтобы увидеть, как такой протокол обеспечивает когерентность, рассмотрим операцию записи, вслед за которой следует операция чтения другим процессором. Поскольку запись требует исключительного права доступа, любая копия, поддерживаемая читающим процессором, должна быть аннулирована (в соответствии с названием протокола).

Таким образом, когда возникает операция чтения, произойдет промах кэш-памяти, который вынуждает выполнить выборку новой копии данных. Для выполнения операции записи мы можем потребовать, чтобы процессор имел достоверную (valid) копию данных в своей кэш-памяти прежде, чем выполнять в нее запись.

Таким образом, если оба процессора попытаются записать в один и тот же элемент данных одновременно, один из них выиграет состязание у второго (мы вскоре увидим, как принять решение, кто из них выиграет) и вызывает аннулирование его копии. Другой процессор для завершения своей операции записи должен сначала получить новую копию данных, которая теперь уже должна содержать обновленное значение.

Альтернативой протоколу записи с аннулированием является обновление всех копий элемента данных в случае записи в этот элемент данных. Этот тип протокола называется протоколом записи с обновлением (write update protocol) или протоколом записи с трансляцией (write broadcast protocol). Обычно в этом протоколе для снижения требований к полосе пропускания полезно отслеживать, является ли слово в кэш-памяти разделяемым объектом, или нет, а именно, содержится ли оно в других кэшах. Если нет, то нет никакой необходимости обновлять другой кэш или транслировать в него обновленные данные.

Разница в производительности между протоколами записи с обновлением и с аннулированием определяется тремя характеристиками:

1. Несколько последовательных операций записи в одно и то же слово, не перемежающихся операциями чтения, требуют нескольких операций трансляции при использовании протокола записи с обновлением, но только одной начальной операции аннулирования при использовании протокола записи с аннулированием.

2. При наличии многословных блоков в кэш-памяти каждое слово, записываемое в блок кэша, требует трансляции при использовании протокола записи с обновлением, в то время как только первая запись в любое слово блока нуждается в генерации операции аннулирования при использовании протокола записи с аннулированием. Протокол записи с аннулированием работает на уровне блоков кэш-памяти, в то время как протокол записи с обновлением должен работать на уровне отдельных слов (или байтов, если выполняется запись байта).

3. Задержка между записью слова в одном процессоре и чтением записанного значения другим процессором обычно меньше при использовании схемы записи с обновлением, поскольку записанные данные немедленно транслируются в процессор, выполняющий чтение (предполагается, что этот процессор имеет копию данных). Для сравнения, при использовании протокола записи с аннулированием в процессоре, выполняющим чтение, сначала произойдет аннулирование его копии, затем будет производиться

чение данных и его приостановка до тех пор, пока обновленная копия блока не станет доступной и не вернется в процессор.

Эти две схемы во многом похожи на схемы работы кэш-памяти со сквозной записью и с записью с обратным копированием. Также как и схема задержанной записи с обратным копированием требует меньшей полосы пропускания памяти, так как она использует преимущества операций над целым блоком, протокол записи с аннулированием обычно требует менее тяжелого трафика, чем протокол записи с обновлением, поскольку несколько записей в один и тот же блок кэш-памяти не требуют трансляции каждой записи. При сквозной записи память обновляется почти мгновенно после записи (возможно с некоторой задержкой в буфере записи). Подобным образом при использовании протокола записи с обновлением другие копии обновляются так быстро, насколько это возможно. Наиболее важное отличие в производительности протоколов записи с аннулированием и с обновлением связано с характеристиками прикладных программ и с выбором размера блока.

Основы реализации. Ключевым моментом реализации в многопроцессорных системах с небольшим числом процессоров как схемы записи с аннулированием, так и схемы записи с обновлением данных, является использование для выполнения этих операций механизма шины. Для выполнения операции обновления или аннулирования процессор просто захватывает шину и транслирует по ней адрес, по которому должно производиться обновление или аннулирование данных.

Все процессоры непрерывно наблюдают за шиной, контролируя появляющиеся на ней адреса. Процессоры проверяют не находится ли в их кэш-памяти адрес, появившийся на шине. Если это так, то соответствующие данные в кэше либо аннулируются, либо обновляются в зависимости от используемого протокола. Последовательный порядок обращений, присущий шине, обеспечивает также строго последовательное выполнение операций записи, поскольку когда два процессора конкурируют за выполнение записи в одну и ту же ячейку, один из них должен получить доступ к шине раньше другого.

Один процессор, получив доступ к шине, вызовет необходимость обновления или аннулирования копий в других процессорах. В любом случае, все записи будут выполняться строго последовательно. Один из выводов, который следует сделать из анализа этой схемы, заключается в том, что запись в разделяемый элемент данных не может закончиться до тех пор, пока она не захватит доступ к шине.

В дополнение к аннулированию или обновлению соответствующих копий блока кэш-памяти, в который производилась запись, мы должны также разместить элемент данных, если при записи происходит промах кэш-памяти. В кэш-памяти со сквозной записью последнее значение элемента данных найти легко, поскольку все записываемые данные всегда посылаются также и в память, из которой последнее записанное значение элемента данных может быть выбрано (наличие буферов записи может привести к некоторому усложнению).

Однако для кэш-памяти с обратным копированием задача нахождения последнего значения элемента данных сложнее, поскольку это значение скорее всего находится в кэш, а не в памяти. В этом случае используется та же самая схема наблюдения, что и при записи: каждый процессор наблюдает и контролирует адреса, помещаемые на шину. Если процессор обнаруживает, что он имеет модифицированную ("грязную") копию блока кэш-памяти, то именно он должен обеспечить пересылку этого блока в ответ на запрос чтения и вызвать отмену обращения к основной памяти. Поскольку кэш с обратным копированием предъявляют меньшие требования к полосе пропускания памяти, они намного предпочтительнее в мультипроцессорах, несмотря на некоторое увеличение сложности. Поэтому далее мы рассмотрим вопросы реализации кэш-памяти с обратным копированием.

Для реализации процесса наблюдения могут быть использованы обычные теги кэш. Более того, упоминавшийся ранее бит достоверности (*valid bit*), позволяющий легко реализовать аннулирование. Промахи операций чтения, вызванные либо аннулированием, либо каким-нибудь другим событием, также не сложны для понимания, поскольку они просто основаны на возможности наблюдения. Для операций записи мы хотели бы также знать, име-

ются ли другие кэшированные копии блока, поскольку в случае отсутствия таких копий, запись можно не посылать на шину, что сокращает время на выполнение записи, а также требуемую полосу пропускания.

Чтобы отследить, является ли блок разделяемым, мы можем ввести дополнительный бит состояния (shared), связанный с каждым блоком, точно также как это делалось для битов достоверности (valid) и модификации (modified или dirty) блока. Добавив бит состояния, определяющий является ли блок разделяемым, мы можем решить вопрос о том, должна ли запись генерировать операцию аннулирования в протоколе с аннулированием, или операцию трансляции при использовании протокола с обновлением. Если происходит запись в блок, находящийся в состоянии "разделяемый" при использовании протокола записи с аннулированием, кэш формирует на шине операцию аннулирования и помечает блок как частный (private). Никаких последующих операций аннулирования этого блока данный процессор посылать больше не будет. Процессор с исключительной (exclusive) копией блока кэш-памяти обычно называется "владельцем" (owner) блока кэш-памяти.

При использовании протокола записи с обновлением, если блок находится в состоянии "разделяемый", то каждая запись в этот блок должна транслироваться. В случае протокола с аннулированием, когда посылается операция аннулирования, состояние блока меняется с "разделяемый" на "неразделяемый" (или "частный"). Позже, если другой процессор запросит этот блок, состояние снова должно измениться на "разделяемый". Поскольку наш наблюдающий кэш видит также все промахи, он знает, когда этот блок кэша запрашивается другим процессором, и его состояние должно стать "разделяемый".

Поскольку любая транзакция на шине контролирует адресные теги кэша, потенциально это может приводить к конфликтам с обращениями к кэшу со стороны процессора. Число таких потенциальных конфликтов можно снизить применением одного из двух методов: дублированием тегов, или использованием многоуровневых кэшей с "охватом" (inclusion), в которых уровни, находящиеся ближе к процессору являются поднабором уровней, находящихся дальше от него. Если теги дублируются, то обращения

процессора и наблюдение за шиной могут выполняться параллельно. Конечно, если при обращении процессора происходит промах, он должен будет выполнять арбитраж с механизмом наблюдения для обновления обоих наборов тегов.

Точно также, если механизм наблюдения за шиной находит совпадающий тег, ему будет нужно проводить арбитраж и обращаться к обоим наборам тегов кэш (для выполнения аннулирования или обновления бита "разделяемый"), возможно также и к массиву данных в кэше, для нахождения копии блока. Таким образом, при использовании схемы дублирования тегов процессор должен приостановиться только в том случае, если он выполняет обращение к кэш в тот же самый момент времени, когда механизм наблюдения обнаружил копию в кэш. Более того, активность механизма наблюдения задерживается только тогда, когда кэш имеет дело с промахом.

Если процессор использует многоуровневый кэш со свойствами охвата, тогда каждая строка в основном кэш имеется и во вторичном кэш. Таким образом, активность по наблюдению может быть связана с кэш второго уровня, в то время как большинство активностей процессора могут быть связаны с первичным кэш. Если механизм наблюдения получает попадание во вторичный кэш, тогда он должен выполнять арбитраж за первичный кэш, чтобы обновить состояние и возможно найти данные, что обычно будет приводить к приостановке процессора. Такое решение было принято во многих современных системах, поскольку многоуровневый кэш позволяет существенно снизить требований к полосе пропускания. Иногда может быть даже полезно дублировать теги во вторичном кэш, чтобы еще больше сократить количество конфликтов между активностями процессора и механизма наблюдения.

В реальных системах существует много вариаций схем когерентности кэш, в зависимости от того используется ли схема на основе аннулирования или обновления, построена ли кэш-память на принципах сквозной или обратной записи, когда происходит обновление, а также имеет ли место состояние "владения" и как оно реализуется. В таблице 8 представлены несколько протоколов

с наблюдением и некоторые машины, которые используют эти протоколы.

Таблица 8

Наименование	Тип протокола	Стратегия записи в память	Уникальные свойства Применение
Одиночная запись	Запись с аннулированием	Обратное копирование при первой записи	Первый описанный в литературе протокол наблюдения -
Synapse N+1	Запись с аннулированием	Обратное копирование	Точное состояние, где "владельцем является память" Машины Synapse. Первые машины с когерентной кэш-памятью
Berkely	Запись с аннулированием	Обратное копирование	Состояние "разделяемый" Машина SPUR университета Berkely
Illinois	Запись с аннулированием	Обратное копирование	Состояние "приватный"; может передавать данные из любого кэша Серии Power и Challenge компании Silicon Graphics
"Firefly"	Запись с трансляцией	Обратное копирование для "приватных" блоков и сквозная запись для "разделяемых"	Обновление памяти во время трансляции SPARCcenter 2000

9.5. Многопроцессорные системы с локальной памятью

Существуют два различных способа построения крупномасштабных систем с распределенной (локальной) памятью. Простейший способ заключается в том, чтобы исключить аппаратные механизмы, обеспечивающие когерентность кэш-памяти, и сосредоточить внимание на создании масштабируемой системы памяти.

Наиболее известным примером такой системы является компьютер T3D компании Cray Research. В этих машинах память распределяется между узлами (процессорными элементами) и все узлы соединяются между собой посредством того или иного типа

сети. Доступ к памяти может быть локальным или удаленным. Специальные контроллеры, размещаемые в узлах сети, могут на основе анализа адреса обращения принять решение о том, находятся ли требуемые данные в локальной памяти данного узла, или размещаются в памяти удаленного узла. В последнем случае контроллеру удаленной памяти посылается сообщение для обращения к требуемым данным.

Чтобы обойти проблемы когерентности, разделяемые (общие) данные не кэшируются. Конечно, с помощью программного обеспечения можно реализовать некоторую схему кэширования разделяемых данных путем их копирования из общего адресного пространства в локальную память конкретного узла. В этом случае когерентностью памяти также будет управлять программное обеспечение. Преимуществом такого подхода является практически минимальная необходимая поддержка со стороны аппаратуры, хотя наличие, например, таких возможностей как блочное (групповое) копирование данных было бы весьма полезным. Недостатком такой организации является то, что механизмы программной поддержки когерентности подобного рода кэш-памяти компилятором весьма ограничены. Существующая в настоящее время методика в основном подходит для программ с хорошо структурированным параллелизмом на уровне программного цикла.

Машины с архитектурой, подобной Cray T3D, называют процессорами (машинами) с массовым параллелизмом (MPP - Massively Parallel Processor). К машинам с массовым параллелизмом предъявляются взаимно исключающие требования. Чем больше объем устройства, тем большее число процессоров можно расположить в нем, тем длиннее каналы передачи управления и данных, а значит и меньше тактовая частота. Произошедшее возрастание нормы массивности для больших машин до 512 и даже 64К процессоров обусловлено не ростом размеров машины, а повышением степени интеграции схем, позволившей за последние годы резко повысить плотность размещения элементов в устройствах. Топология сети обмена между процессорами в такого рода системах может быть различной. В таблице 9 приведены характеристики сети обмена для некоторых коммерческих MPP.

Таблица 9

Фирма	Название	Количество узлов	Базовая топология
Thinking Machines	CM-2	1024-4096	12-мерный куб
nCube	nCube/ten	1-1024	10-мерный куб
Intel	iPSC/2	16-128	7-мерный куб
Maspar	MP-1216	32-512	2-мерная сеть+ступенчатая Omega
Intel	Delta	540	2-мерная сеть
Thinking Machines	CM-5	32-2048	многоступенчатое толстое дерево
Meiko	CS-2	2-1024	многоступенчатое толстое дерево
Intel	Paragon	4-1024	2-мерная сеть
Cray Research	T3D	16-1024	3-мерный тор

Для построения крупномасштабных систем альтернативой рассмотренному в предыдущем разделе протоколу наблюдения может служить протокол на основе справочника, который отслеживает состояние кэшей. Такой подход предполагает, что логически единый справочник хранит состояние каждого блока памяти, который может кэшироваться. В справочнике обычно содержится информация о том, в каких кэш имеются копии данного блока, модифицировался ли данный блок и т.д. В существующих реализациях этого направления справочник размещается рядом с памятью.

Имеются также протоколы, в которых часть информации размещается в кэш-памяти. Положительной стороной хранения всей информации в едином справочнике является простота протокола, связанная с тем, что вся необходимая информация сосредоточена в одном месте. Недостатком такого рода справочников является его размер, который пропорционален общему объему памяти, а не размеру кэш-памяти. Это не составляет проблемы для машин, состоящих, например, из нескольких сотен процессоров, поскольку связанные с реализацией такого справочника накладные расходы можно преодолеть. Но для машин большего размера необходима методика, позволяющая эффективно масштабировать структуру справочника.

В частности, чтобы предотвратить появление узкого горла в системе, связанного с единым справочником, можно распределить части этого справочника вместе с устройствами распределенной локальной памяти. Таким образом можно добиться того, что обращения к разным справочникам (частям единого справочника) могут выполняться параллельно, точно также как обращения к локальной памяти в распределенной памяти могут выполняться параллельно, существенно увеличивая общую полосу пропускания памяти. В распределенном справочнике сохраняется главное свойство подобных схем, заключающееся в том, что состояние любого разделяемого блока данных всегда находится во вполне определенном известном месте.

На рис. 69 показан общий вид подобного рода машины с распределенной памятью. Вопросы детальной реализации протоколов когерентности памяти для таких машин выходят за рамки настоящего обзора.

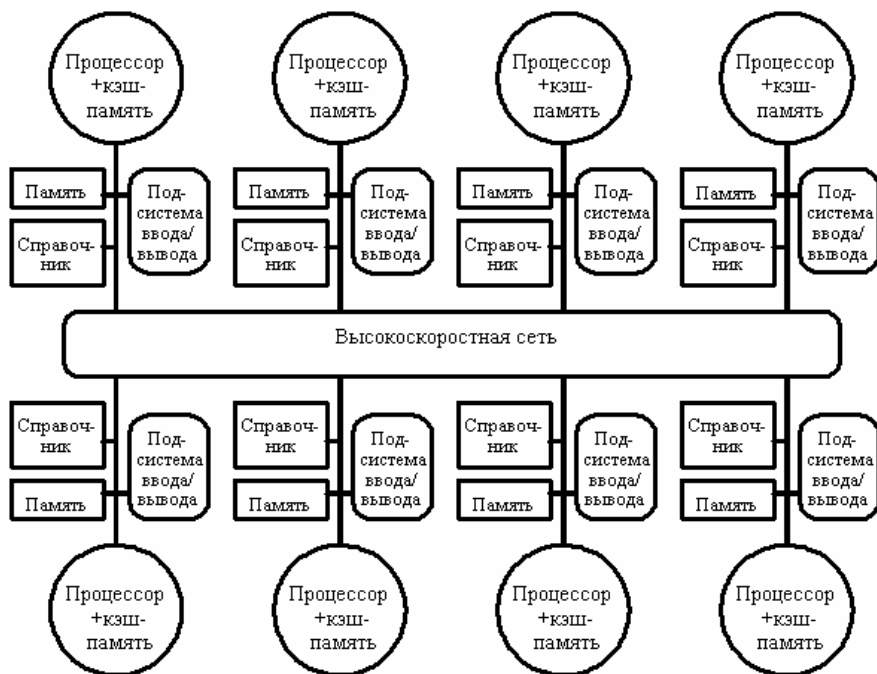


Рис. 69

10. Режимы обмена в МПС

Организация обмена в МПС - важная функция микропроцессора. В обмене принимают участие активное и пассивные устройства. Активным устройством в большинстве случаев является МП, пассивным - основная память или ВУ.

Основными режимами обмена являются программно - управляемый обмен, обмен в режиме прерывания и обмен в режиме прямого доступа к памяти.

Программно-управляемый обмен. Обмен осуществляется по инициативе МП и предназначен для обмена данными между МП и ВУ (или ОП) и их программной обработки. Алгоритм работы МП приведен на рис.70.

Программно - управляемый обмен осуществляется по инициативе обрабатываемой команды и включает чтение информации в микропроцессор из ОП, запись информации в ОП из МП, ввод информации в МП из ВУ и вывод информации из МП во ВУ. Рассмотрим перечисленные виды обмена.

Чтение информации в микропроцессор из основной памяти (рис. 71, а) начинается с момента выдачи из МП на ША значения адреса ячейки ОП, из которой должно быть произведено чтение информации. По синхронизирующему импульсу «чтение» (RD), поступающему из МП на ШУ, активизируются искомые ячейки ОП. Информация из ОП поступает на ШД, передается в МП и записывается в соответствующий регистр МП.

Запись информации в основную память из МП (рис. 71,б) начинается так же, как и в первом случае: из МП на ША поступает значение адреса ячейки ОП, в которую должна быть произведена запись, а МП вырабатывает на линии ШУ сигнал «запись» (WR). Одновременно информация из МП поступает на ШД, передается в ОП и записывается в соответствующую ячейку памяти.

Ввод информации в МП из внешнего устройства начинается по сигналу синхронизации от управляющего устройства или МП, но на ША поступает адрес конкретного канала КВВ, который соединен с требуемым ВУ и через который будет происходить ввод (чтение) информации в МП. Такой канал называется портом. Через некоторое время на линии ШУ МП формирует управляющий

сигнал RD «чтение» (или «ввод»). Запрошенный по указанному адресу порт активизируется, и по сигналу RD информация из ВУ поступает через порт на ШД. По ней информация передается в МП.

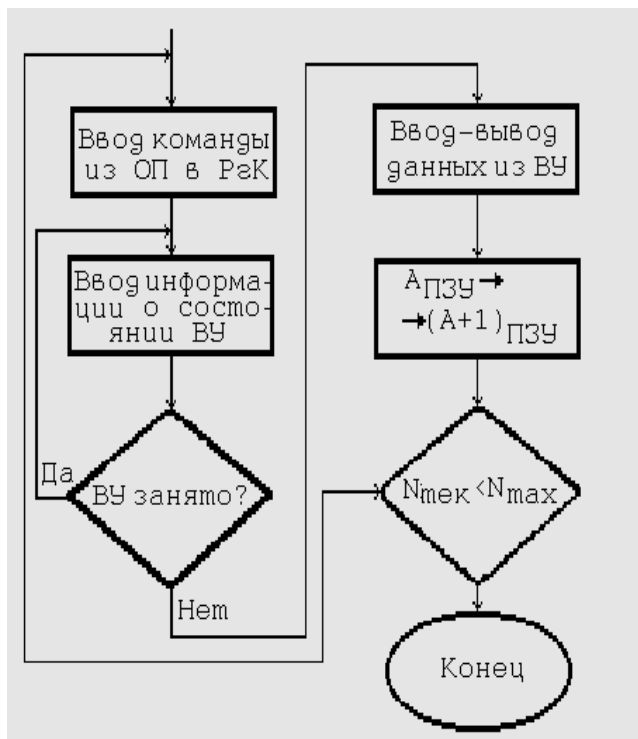


Рис. 70.

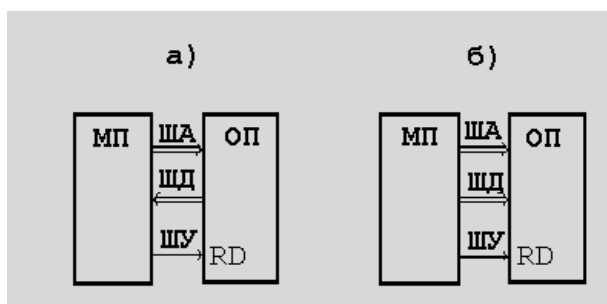


Рис. 71.

Вывод информации из МП во внешнее устройство осуществляется путем формирования МП на линиях ША адреса канала (порта) КВВ, который соединен с требуемым ВУ. Через некоторый промежуток времени МП формирует на линии ШУ сигнал WR «запись» (или «вывод») и выдает на ШД информацию, которая должна быть записана (выведена) в требуемое ВУ. Запрошенный по указанному адресу порт вывода активизируется и информация с ШД поступает в заданное ВУ.

Необходимо отметить, что в рассмотренных режимах обмена всегда участвует специальный регистр МП, называемый аккумулятором: из него информация передается в ШД при выводе ее из МП, и в него она поступает при вводе информации в МП.

При рассмотрении описанных выше режимов обмена с участием МП не ставился вопрос о готовности пассивных ВУ к проведению обмена. Между тем их быстродействие существенно различается в зависимости от вида ВУ. Если в МПС используются ВУ, имеющие быстродействие, сравнимое с быстродействием МП, то МП как правило не проводит анализ готовности ВУ к обмену. Обмен в этом случае носит название синхронным обменом. В тех случаях, когда быстродействие ВУ ниже быстродействия МП (например, клавиатура пульта оператора), то синхронный способ обмена неприменим и используется так называемый асинхронный обмен.

Асинхронный обмен происходит также под управлением программы, но лишь в том случае, когда ВУ подготовлено к обмену. Об этом сообщает сигнал готовности, формируемый ВУ через КВВ (или самим КВВ) на соответствующей линии ШУ. Процесс обмена, инициированный программой, начинается с анализа процессором готовности ВУ к обмену. При отсутствии сигнала готовности МП переходит в состояние ожидания, о чем извещает остальные функциональные модули специальным сигналом (WALT) на одной из линий ШУ. После прихода сигнала готовности происходит непосредственная процедура обмена.

Основным недостатком асинхронного обмена являются потери времени процессора на ожидание того момента, когда устройство будет готово к обмену, и такие потери для некоторых устройств могут оказаться значительными. Так, например, при вводе

информации с пульта оператора среднее время между нажатиями клавиши составляет не менее 0,1 с. Время же самой операции ввода информации с клавиши в МП обычно не превышает 10 мкс. Очевидно, что полезное время работы МП в этом случае не превышает 0,01% общего времени обмена.

Из приведенного примера видно, что за время ожидания очередного сигнала готовности МП способен выполнить достаточно большое число операций в соответствии с командами программы, если бы сигнал готовности ВУ мог останавливать (прерывать) выполнение основной программы и переводить МП в режим выполнения процедуры обмена. Сигнал готовности такого вида называется сигналом прерывания, а способ обмена с использованием сигналов прерывания получил название обмена в режиме прерывания.

Обмен в режиме прерывания. Обмен в режиме прерывания предназначен для обработки программ обслуживания запросов прерывания, сформированных ВУ в процессе накопления ими информации за время работы. Время формирования запросов прерывания ВУ - явление случайное и, в большинстве случаев, не может быть запрограммировано.

Обмен в режиме прерывания производится по инициативе ВУ или КВВ, обслуживающего данное ВУ, и осуществляется именно в те моменты времени, когда соответствующее ВУ готово к передаче данных в МП. По мере готовности к передаче данных контроллер прерываний, обслуживающий данное ВУ, вырабатывает сигнал запроса прерывания, который МП анализирует и, при необходимости, прерывает обрабатываемую программу и переходит к операции обмена - вводу и обработке программы обслуживания прерывания.

Различают прерывания аппаратные, программные и специальные.

Аппаратные прерывания, на практике еще называемые внешними прерываниями, имеют место при воздействии сигналов, которые вырабатываются ВУ, требующими обслуживания. Аппаратные прерывания используются, как правило, для обслуживания ВУ по запросу этих устройств. Они могут быть немаскируемые и маскируемые.

Немаскируемые прерывания - прерывания, которые вызываются внешними, аппаратными средствами и не могут быть запрещены выполняемой программой. Запросы на такие прерывания подаются на специальный вход микропроцессора - вход немаскируемых прерываний. Они обслуживаются обязательно и немедленно вне зависимости от важности выполняемой в данный момент времени программы.

Маскируемые прерывания - прерывания, которые могут быть разрешены или запрещены программным путем - включением в программу специальных команд, разрешающих или запрещающих прерывания на данном участке программы. Для реализации таких прерываний в микропроцессоре имеется один или несколько входов для запросов на обслуживание маскируемых прерываний.

Программные прерывания происходят под воздействием команд прерывания, включенных в основную программу. Здесь инициатива программного прерывания исходит от самой программы. Программные прерывания используются:

- для обслуживания устройств ввода - вывода по опросу,
- для вызова вспомогательных программ операционной системы - так называемых "утилит".

В последнем случае механизм прерываний оказывается более эффективным, чем механизм перехода на программу обслуживания прерывания, хотя в принципе они очень близки.

Специальные прерывания возникают в системе в ходе выполнения основной программы под воздействием сигналов, вырабатываемых внутренними аппаратными средствами. Специальные прерывания можно назвать также внутренними прерываниями. Причинами их появления могут быть:

- а) программные сбои или ошибки, являющиеся следствием попыток выполнить неразрешенную (неверную) команду или обратиться к запрещенной области памяти (при ошибке адресации), и др.;
- б) аппаратные сбои, являющиеся следствием критического изменения или внезапного отключения питания, неисправностей отдельных узлов аппаратуры, вызывающих неправильное функционирование аппаратуры, и др.;

в) переполнение разрядной сетки, т. е. получение в ходе вычислений чисел, выходящих за пределы диапазона допустимых значений;

г) трассировка - выполнение программы в пошаговом режиме при ее отладке и некоторые другие.

В МПС используются одноуровневые и многоуровневые системы обработки прерывания.

При одноуровневой системе обработки прерываний все контроллеры прерывания имеют одинаковый приоритет по отношению к приоритету системы и подключаются последовательно к линии, по которой передается сигнал разрешения прерывания (рис. 72). При этом ближайшему к МП контроллеру прерываний (KPr_1) присваивается наименьший адрес, а самому удаленному (KPr_N) - наибольший.

Запросы от ВУ через соответствующие контроллеры прерываний KPr поступают на вход «1» запроса прерывания МП по единственной линии, к которой подключаются все KPr системы. При этом запросы на прерывание от одного или нескольких KPr могут маскироваться программно. Если на линии появляется запрос от одного какого либо KPr , то МП воспринимает его, анализирует на предмет приоритетности ВУ, пославшего запрос, и, в случае более высокого приоритета устройства по сравнению с приоритетом системы, проводит дополнительные операции по сохранению обрабатываемой программы и, после этого, выдает на выходе «2» сигнал разрешения прерывания.

На практике возможно одновременное поступление запросов на линию «1» от двух или более KPr . Если приоритет МП, ведущего обработку текущей программы, выше приоритета устройств, то любой запрос прерывания им игнорируется. Если же приоритет МП ниже приоритета всех ВУ, то в ответ на любой запрос прерывания МП подготавливается к переходу обработки программы обслуживания прерываний и посылает сигнал на линию «2» разрешения прерывания. Этот сигнал последовательно проходит через все контроллеры прерываний, начиная с самого ближнего по линии связи к МП, и кончая тем, который послал сигнал прерывания. Этот KPr запрещает, или блокирует, дальнейшее его распространение и выдает на ШД начальный адрес программы об-

служивания прерывания, уникальный для данного ВУ. МП воспринимает этот адрес, транслирует его передачу на ША, вводит первую команду программы обслуживания прерывания и начинает ее выполнение.

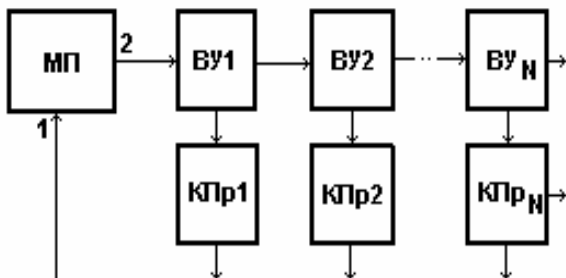


Рис.72.

Последовательность действий, выполняемая МП при обработке прерывания от ВУ, которая справедлива также при обработке и других классов прерываний, состоит в следующем.

1. Сигнал прерывания поступает в МП во время выполнения $(i - 1)$ - й команды обрабатываемой программы. Микропроцессор полностью завершает эту команду, при этом в счетчике команд уже подготовлен адрес следующей i - й команды. По специальной команде, с помощью указателя стека, МП запоминает в стеке сначала содержимое счетчика команд (т.е. адрес i - й команды обрабатываемой программы), а затем текущее содержимое регистра состояния МП, которое считается «старым» словом состояния (ССП) МП, и, при необходимости, содержимое программно - доступных регистров (например, аккумулятора, РОН), участвовавших в формировании результатов обработки прерванной программы, а также дополнительно проводит необходимые операции по обслуживанию прерывания, зависящий от источника прерывания.

2. По адресу «вектора прерывания», сформированного контроллером прерываний, МП загружает из вектора прерывания в счетчик команд начальный адрес программы обслуживания прерывания, а в регистр состояний - «новое» слово состояния МП.

Вектор прерывания - это адрес первой команды программы обслуживания определенного типа прерываний. В некоторых микропроцессорах, например, в МП Z80000 фирмы Zilog (США), K1801BM3, вектор включает в себя не только адрес первой команды программы обслуживания прерывания, но и слово состояния процессора, в котором отведено 2, 3 или 4 разряда для указания уровня приоритета данного прерывания.

3. МП переходит к запуску и обработке программы обслуживания прерывания.

В «новом» ССП как правило устанавливается новое значение приоритета МП по отношению ко всем или определенному ВУ системы. Оно определяет возможность повторного прерывания МП от того же или другого источника во время выполнения программы обслуживания прерывания. Если есть необходимость в запрете прерывания до полного завершения программы обслуживания прерывания, то в «новом» слове состояния МП на время обработки программы обслуживания прерывания должен быть установлен высший приоритет МП.

После своего завершения программа обслуживания прерывания специальной командой восстанавливает из стека записанное туда ранее содержимое регистров, соответствующее прерванной программе. Затем эта же программа выполняет команду выхода из прерывания, в результате которой адрес i -й команды, «старое» слово состояния МП и, если это необходимо, содержимое программно - доступных регистров МП будут в него загружены. С этого момента прерванная программа продолжит свою работу с i -й команды прерванной программы.

Многоуровневая система прерывания характеризуется тем, что в системе может быть несколько КПр, которые могут одновременно послать запросы от ВУ на обслуживание прерывания. Очевидно, в МП должен быть предусмотрен механизм для приема в каждый момент времени единственного запроса для обслуживания прерывания. Этот механизм реализуется, как правило, аппаратными средствами и позволяет в первую очередь выбирать порядок обслуживания ВУ: либо первым удовлетворяется запрос прерывания от того контроллера, который включен ближе к МП в цепочке передачи сигналов «запрос прерывания - разрешение пре-

рывания», либо первым удовлетворяется запрос прерывания с КПр, программно имеющего наивысший приоритет.

Разделение всех ВУ в системе по уровням приоритетности обуславливается важностью формируемой ими информации. ВУ, в которых накопленная информация требует незамедлительной ее обработки, имеют наивысший или более высокий приоритет по отношению к МП. Те ВУ, в которых информация может храниться некоторое время без обработки, имеют средний или более низкий приоритет.

Максимальное число программ, обслуживающих ВУ, которые могут прервать друг друга, характеризуют «уровень прерывания». При этом порядок обработки программ при одновременном их запросе определяется уровнем приоритета обслуживаемых ВУ.

В многоуровневых системах прерывания задание уровня приоритета осуществляется тремя способами:

а) использованием в самом МП (или в контроллере прерываний) нескольких входов запросов на прерывание, каждый из которых имеет свой уровень приоритета;

б) с помощью 2 - 4-разрядной шины для подачи кода, несущего информацию об уровне приоритета того или иного КПр, запросившего прерывание;

в) использованием внешних аппаратных средств.

В многоуровневых системах прерываний программа обслуживания ВУ низкого уровня может быть прервана запросом на прерывание от Кпр, обслуживающего ВУ высокого уровня. Если же запрос на прерывание от КПр имеет тот же уровень, что и обслуживаемое на момент запроса прерывание, или более низкий, то МП не реагирует на этот запрос до тех пор, пока не закончит обслуживание устройства. Здесь условно показано, что после поступления сигнала запроса прерывания управление запоминанием состояния и возвратом в прерываемую программу возложено на саму прерывающую программу. В этом случае прерывающая программа состоит из трех частей:

— подготовительной, осуществляющей запоминание состояния прерванной программы,

— заключительной, обеспечивающей восстановление состояния прерванной программы,

— прерывающей программы, выполняющей затребованный запросом режим. с более высоким уровнем приоритета.

Для оценки эффективности системы прерывания, реализующей обмен, используются следующие характеристики системы прерывания.

Время реакции - время между появлением запроса прерывания и началом выполнения прерывающей программы.

На рис. 73 приведена упрощенная временная диаграмма процесса прерывания.

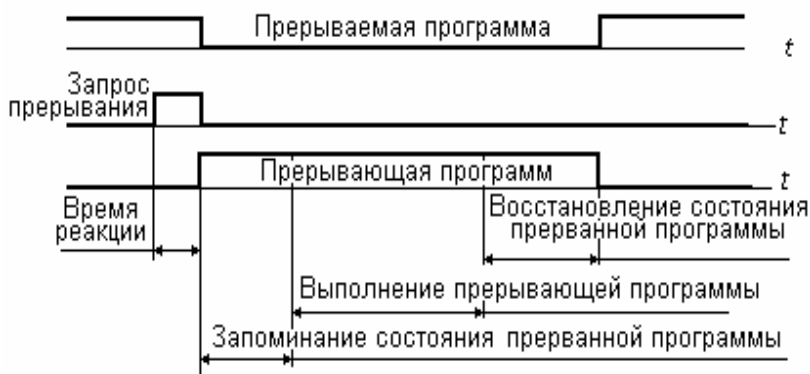


Рис. 73

Для одного и того же запроса прерывания задержки в исполнении прерывающей программы зависят от того, сколько программ с более высоким приоритетом ВУ ждут обслуживания. Поэтому время реакции определяется для запроса с наивысшим приоритетом ВУ.

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. Оно может оказаться недопустимо большим для систем, работающих в реальном масштабе времени. В таких системах часто допускается прерывание обрабатываемой программы после любого такта выполнения команды. Однако при этом возрастает количество информации, подлежащей запоминанию и восстановлению при пере-

ключении программ, так как в этом случае необходимо сохранять также и состояния в момент прерывания счетчика тактов, регистра кода операции и некоторых других. Поэтому такая организация прерывания возможна в МПС 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 влияния на следующие такты работы системы.

Затраты времени на переключение программ равны суммарному расходу времени на запоминание и восстановление состояния программы.

Глубина прерывания - максимальное число программ, которые могут прерывать друг друга. Если после перехода к прерывающей программе и вплоть до ее окончания прием других запросов запрещается, то система имеет глубину прерывания, равную «1». Если допускается последовательное прерывание до N программ, то глубина прерывания будет равна N .

Глубина прерывания обычно совпадает с числом уровней приоритета в системе прерываний. На рис. 74 показан обмен в режиме прерывания в системах с единичной глубиной прерывания (а) и с различной глубиной прерывания (б) в предположении, что приоритет каждого следующего запроса выше предыдущего.



Рис. 74

В первом случае появление каждого следующего сигнала «2», «3» и т. д. запроса прерывания не влияет на алгоритм работы системы - МП продолжает выполнение программы «1», «2» обработки предыдущего прерывания до ее завершения. Только после этого он приступает к обработке следующей программы обработки прерывания.

Во втором случае обработка текущей программы прерывается в случае, если на вход системы поступает сигнал запроса прерывания от устройства с более высоким приоритетом, чем приоритет текущей программы. Так, например, сигнал запроса прерывания «2» (рис. 74,б) ВУ более высокого приоритета, чем приоритет МПС, ведет к прерыванию текущей программы и началу обработки программы обслуживания прерывания ВУ, выставившего запрос «2». Аналогично обслуживается сигнал «3» от ВУ, имеющего более высокий приоритет, чем ВУ, пославшего сигнал «2». После обработки программ «3» и «2» МП возвращается в ранее прерванную программу «1».

Необходимо отметить, что, чем большее значение глубины прерывания имеет система, тем более быстрой реакцией на срочные запросы она обладает. Однако, если запрос окажется необслуженным к моменту прихода нового запроса от того же источника, то возникает так называемое насыщение системы прерывания. В этом случае предыдущий запрос прерывания от данного источника будет системой утрачен, что является недопустимым. Для устранения этого явления быстроедействие МПС, характеристики системы прерывания, число источников прерывания и частота возникновения запросов должны быть согласованы таким образом, чтобы насыщение было невозможным.

Приоритетное обслуживание запросов прерывания. Вектор прерывания 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. Главное место в процедуре перехода к прерывающей программе занимают передача из соответствующего регистра (регистров)

процессора в память (в частности, в 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 процессом.

Процедура организации перехода к прерывающей программе включает в себя выделение из выставленных запросов такого, который имеет наибольший приоритет.

Различают абсолютный и относительный приоритеты. Запрос, имеющий абсолютный приоритет, прерывает выполняемую программу и инициирует выполнение соответствующей прерывающей программы. Запрос с относительным приоритетом является первым кандидатом на обслуживание после завершения выполнения текущей программы. Если наиболее приоритетный из выставленных запросов прерывания не превосходит по уровню приоритета выполняемую процессором программу, то запрос прерывания игнорируется или его обслуживание откладывается до завершения выполнения текущей программы.

Простейший способ установления приоритетных соотношений между запросами (уровнями) прерывания состоит в том, что приоритет определяется порядком присоединения линии сигналов запросов ко входам системы прерывания. При появлении нескольких запросов прерывания первым воспринимается запрос, поступивший на вход с меньшим номером. В этом случае приоритет является жестко фиксированным. Изменить приоритетные соотношения в этом случае можно лишь пересоединением линий сигналов запросов на входах системы прерывания.

Процедура прерывания с опросом источников (флажков) прерывания. При указанном способе задания приоритета между запросами каждому источнику запросов соответствует разряд (флажок) в регистре запросов прерывания (регистре флажков). При наличии запроса или нескольких запросов прерывания формируется сигнал подтверждения прерывания, инициирующий выполняемую специальной программой или аппаратурой процедуру опроса регистра запросов прерывания для установления источника, выставившего запрос прерывания наивысшего приоритета.

Более гибким и динамичным является векторное прерывание, при котором исключается опрос источников прерывания.

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 надобности приоритетные соотношения программным путем, то есть приоритет между прерывающими программами должен быть динамичным или, другими словами, программно - управляемым.

В МПС программно - управляемый приоритет прерывающих программ может быть реализован по порогу прерывания и по маске прерывания.

Установка приоритета по порогу прерывания предполагает в ходе вычислительного процесса программным путем изменять уровень приоритета (порог прерывания) процессора (а следовательно, и обрабатываемой в данный момент процессором программы) относительно приоритетов запросов источников прерывания (в основном внешних устройств). Порог прерывания задается командой программы специальным кодом порога прерывания, который служит для выделения наиболее приоритетного запроса прерывания, сравнения его приоритета с порогом прерывания и, если он оказывается выше порога, выработки общего сигнала прерывания, по которому начинается процедура прерывания.

Маска прерывания представляет собой двоичный код, разряды которого поставлены в соответствие запросам или классам прерывания. Маска загружается командой текущей программы в регистр маски. Состояние «1» в данном разряде регистра маски разрешает, а состояние «0» запрещает (маскирует) прерывание текущей программы от соответствующего запроса. 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 сигналы логической «1» уста-

навливаются в разряды, соответствующие запросам (прерывающим программами) с более высоким, чем у данной программы, приоритетом.

Обмен в режиме прямого доступа к памяти. В рассмотренных ранее режимах обмен информацией осуществляется между МП и ОП или между МП и ВУ. Однако на практике часто возникает необходимость оперативного обмена информацией между ВУ и ОП без ее обработки. В этом случае при использовании ранее описанных режимов процедура обмена должна содержать два цикла. В первом цикле информация сначала должна быть передана из ВУ (или ОП) в аккумулятор МП, во втором цикле - информация из аккумулятора должна быть занесена в ОП(или ВУ).

При обмене с медленнодействующими ВУ и передачах больших массивов информации такая двухступенчатая процедура существенно снижает скорость обмена, то есть ведет к снижению быстродействия МПС в целом. В связи с этим используется метод обмена, при котором запись информации в ОП из ВУ или считывание информации из памяти во внешнее устройство происходит непосредственно без участия МП. Такой вид обмена получил название обмена в режиме прямого доступа к памяти (ПДП).

Для организации обмена применяется специальное управляющее устройство - контроллер ПДП, который при обмене выполняет функции активного устройства, то есть устанавливает адрес ячейки ОП или порта ВУ, участвующих в обмене, на линиях ША, формирует необходимые управляющие сигналы на линиях ШУ, определяет начало передачи информации по линиям ШД.

При программно - управляемом обмене и при обмене в режиме прерывания магистралью (шинами адреса, данных и управления) распоряжается МП. При обмене в режиме ПДП магистраль должна быть передана в распоряжение контроллера ПДП, а МП необходимо отключить от шин. Для этой цели в МП предусмотрен вход специального управляющего сигнала «запрос захвата шин», при поступлении которого после окончания текущего цикла выполнения команды управляющее устройство МП переводит буферные устройства его шин в режим с высоким выходным сопротивлением (высокоимпедансное состояние). МП при этом отключается от магистралей и его управляющее устройство на специ-

альной линии управления формирует сигнал разрешения захвата шин контроллеру ПДП и представляет магистраль в его распоряжение для передачи либо одного слова, либо целого массива информации.

При передаче одного слова контроллер ПДП занимает магистраль для обмена только на один цикл работы управляющего устройства. На следующем цикле ею снова распоряжается МП. В очередной цикл магистраль снова предоставляется контроллеру ПДП и т. д., пока не будут переданы все слова массива. Все время, пока осуществляется обмен в режиме ПДП, на входе МП присутствует сигнал «запрос захвата шин». Это так называемый мультиплексный обмен в режиме ПДП, при котором МП может продолжать выполнение основной программы в режиме разделения времени с процедурой обмена в режиме ПДП. При этом нет необходимости запоминать в стековой памяти ОЗУ содержимое счетчика команд и аккумулятора, так как МП в обмене не участвует.

При передаче массива информации магистраль занимается контроллером ПДП на все время передачи информации и МП останавливает свою работу на время обмена. Такой режим обмена с ПДП называется монопольным и он обладает максимальной скоростью передачи информации.

Для формирования значений адресов ячеек и необходимых управляющих сигналов в составе контроллера ПДП предусмотрено несколько регистров: регистры адреса, в которых находятся начальное и текущее значения адресов, регистр - счетчик передаваемых слов, регистр управления, регистр состояния и т. п. Обычно в составе контроллера ПДП предусматривается несколько каналов для подключения внешних устройств. В этом случае контроллер помимо прочего осуществляет арбитраж запросов на захват шин магистрали с учетом приоритета подключенных к нему ВУ.

Необходимо отметить, что контроллеры ПДП выполняются в виде конструктивно законченных функциональных модулей или в виде БИС, входящих в комплект МП БИС.

11. Каналы передачи информации в МПС

Работа МПС характеризуется интенсивным обменом информацией между их основными частями: МП, ОП, УВВ и мультисистемными средствами. Связь МП с ОП, УВВ требует нескольких каналов передачи информации - интерфейсов. В зависимости от функционального назначения интерфейсные схемы делятся на несколько уровней (рис. 75).

Шинные интерфейсы 1 уровня обеспечивают обмен информацией между всеми (или основными) модулями микропроцессорной системы. Выбор большинства шин этого уровня определяется архитектурой микропроцессора, например интерфейс И-41 (Multibus) используется для МПК серий К580, К1810, межмодульный параллельный интерфейс (МПИ) - для МПК серий К1801/1809, К1811, К581, К5S8 и др.

Наибольшего применения среди шин этого уровня нашли шинные интерфейсы PCI, МПИ, Unibus (Общая шина), И-41 (Multibus), Multibus 11, Futurebus, Fastbus. Они обеспечивают взаимодействие периферийного оборудования, подключенного через соответствующие контроллеры, с вычислительным ядром МПС. В связи с увеличением скоростей работы внешних устройств, постоянным ростом их числа и необходимостью одновременного использования ВУ требования к пропускной способности системной магистрали постоянно растут. Поэтому их число велико и продолжает расти, что отражает, с одной стороны, постоянный рост числа и совершенствование микропроцессоров, а с другой - возникновение все более сложных задач, решаемых ими.

Очевидно, что при сопряжении МП с ОП практически не требуется никаких дополнительных средств, то для сопряжения МП с УВВ требуются специальные устройства, обеспечивающие передачу определенных наборов сигналов. Поэтому шины обмена информацией подключаются не непосредственно к УВВ, а через интерфейсные устройства, структура, принцип работы и технические характеристики которых в сильной степени зависят от совместимости сопрягаемых компонентов.

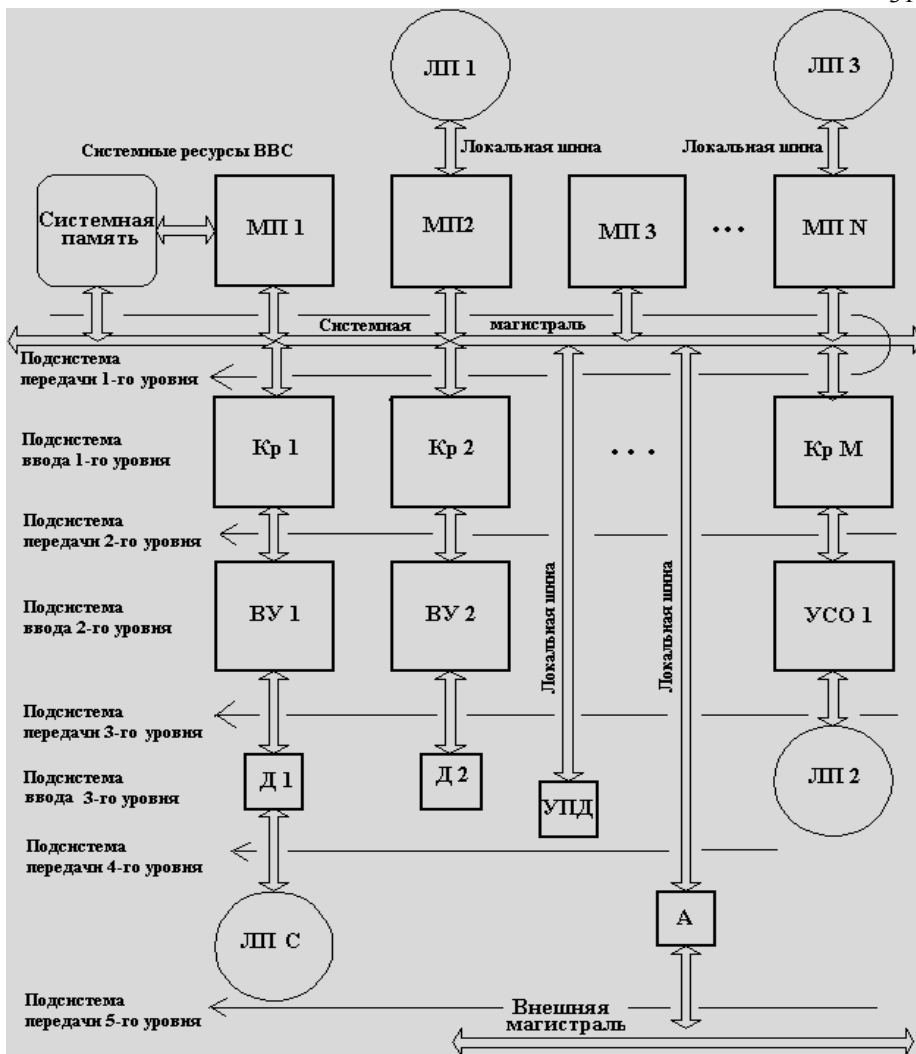


Рис. 75

Совместимость определяется следующими основными признаками: быстродействием, кодами, используемыми для обмена, архитектурой процессора, электрическими характеристиками. Если объединяемые компоненты не соответствуют друг другу по одному или нескольким признакам, то для взаимного подключения используют специальные электронные схемы, называемые интер-

фейсными модулями. Необходимость использования интерфейсных модулей объясняется также и тем, что архитектура процессора с точки зрения набора и организации ШД, ША, набора управляющих сигналов определяет протокол или метод синхронизации МП и УВВ, и все передачи данных, кодов, признаков состояния, управляющих сигналов должны подчиняться этому протоколу обмена. К тому же электрические характеристики МП должны быть совместимы с характеристиками логических схем интерфейса, которые в свою очередь согласуются с ВУ с помощью контроллеров.

Способы структурной и функциональной организации контроллеров ВУ определяются в основном двумя факторами:

- форматами данных и режимами работы конкретных ВУ;
- типом системного интерфейса МПС.

Как показывает практика, создание для конкретного типа ВУ уникального контроллера, обеспечивающего полную электрическую, информационную и конструктивную совместимость данного ВУ с системной магистралью, является сложной с технической и экономической точки зрения задачей. Поэтому наиболее рациональным является стандартизация информационных и управляющих сигналов, которыми обмениваются МП с контроллером и ВУ.

С развитием микроэлектроники появилась возможность реализовать стандартные интерфейсные функции в виде БИС. Для различных МПК БИС разработаны контроллеры, обеспечивающие связь ВУ по стандартному последовательному или по стандартному параллельному каналу передачи данных.

Прежде, чем начать обзор шин, необходимо сказать несколько слов о том, что представляет собой системная шина, и для чего она нужна в компьютере. Шина, в самом простом случае, есть множество проводников для соединения различных компонентов системы в единую систему таким образом, чтобы можно было согласовать их работу. Основной обязанностью системной шины является передача информации между базовым микропроцессором и остальными электронными компонентами системы. По этой шине осуществляется не только передача информации, но и адресация устройств, а также обмен специальными служебными сигналами.

Таким образом, системную шину можно представить как совокупность сигнальных линий, объединенных по их назначению:

- Control lines (управление)
- Address lines (адреса)
- Data lines (данные)

Для того, чтобы описать примерную работу шины, возьмем шину обычного PC, состоящую минимум из линий адреса, данных и линий управления/строба. Самое простое решение, которое здесь можно использовать - это программируемый ввод-вывод. Линии управления используются для синхронизации передачи данных, путем генерирования последовательности импульсов. Возможны две схемы управления, например, отдельные линии управления чтением и записью, либо линия стробирования STROBE и линия чтения - записи в соответствующем состоянии (высокий уровень - для одного сигнала, низкий - для другого).

Шины для PC имеют тенденцию, когда используются отдельные линии управления чтением и записью (фактически две такие линии используются для доступа к памяти, а две дополнительных линии - для осуществления ввода-вывода). В этом случае центральный процессор посылает данные на периферийные устройства, подключенные к шине. ЦП устанавливает стробирующий сигнал по линии ввода - вывода. Этот импульс показывает, что предшествующий адрес на линии адреса правильный, а периферия может начать чтение с шины данных. Кроме перечисленных выше сигналов имеются также и другие сигналы управления, присутствующие на реальной системной шине.

Существует множество системных шин, в том числе и локальных, для PC и других типов компьютеров. Наиболее известными являются S-100, S-100 / IEEE696, ISA, EISA, Nubus, Multibus-II, MCA, Sbus, Mbus, SCSI, VL-Bus, Futurebus+, VME, PCI.

Шина S-100 была создана для 8-разрядных микропроцессоров и различных промышленных приложений. Типичные ее характеристики были такие:

- размеры: 134 мм x 254 мм, 100 выводов
- разъем: 50 выводов на каждой стороне платы
- нерегулируемое напряжение питания: +8В, +16В.

Шина S-100 нашла широкое применение в периферийных платах, она входила в состав плат памяти, устройств последовательного и параллельного интерфейсов, плат контроллеров гибких магнитных дисков, видео-плат, плат музыкальных синтезаторов и т.д. S-100 обеспечивала 16 линий данных, 16 линий адреса (при этом максимальное адресное пространство составляло 64Кбайт), 3 линии питания, 8 линий для прерываний и 39 управляющих линий. Эта шина использовалась для микропроцессоров Intel 8080, Zilog Z-80 и Motorola 6500 и 6800. Некоторые фирмы создали на базе S-100 свои стандарты подобной шины.

Одним из таких примеров может служить стандарт шины S-100/IEEE696, которой разрабатывался в 1983 году. Полученная шина имела следующие характеристики:

- дополнительные 8 разрядов адреса позволили адресовать до 16 Мбайтов памяти (таким образом, всего получилось 24 линии адреса).

- поддержка 16 - разрядных микропроцессоров путем добавления еще двух сигналов sixteen request (SXTR0, 58 линия) и sixteen acknowledge (SIXTN, 60 линия).

- линия 12 была зарезервирована для сигнала немаскируемого прерывания (NMI).

Полная спецификация этой шины включает до 100 сигналов. Рабочая частота при этом достигает 10 МГц. Шина S-100 и ее модификации нашли применение при разработках небольших промышленных приложений. Основными достоинствами этой шины являются низкая цена и поддержка шины большим числом промышленных разработчиков.

У компьютеров IBM PC AT и IBM PC XT системная шина была предназначена для одновременной передачи только 8 разрядов данных, так как используемый в компьютерах микропроцессор i8088 имел 8 линий данных. Кроме этого, системная шина включала 20 адресных линий, которые ограничивали адресное пространство пределом в 1 Мбайт.

Для работы с внешними устройствами в этой шине были предусмотрены также 4 линии аппаратных прерываний и 4 линии для требования внешними устройствами прямого доступа в память (DMA - Direct Memory Access). Для подключения плат расширения

использовались специальные 62-контактные разъемы. Заметим, что системная шина и микропроцессор синхронизировались от одного тактового генератора с частотой 4,77 МГц. Таким образом, теоретически скорость передачи данных могла достигать более 4.5 Мбайт/с.

Шина ISA. В компьютерах PC AT, использующих микропроцессор i80286, впервые стала применяться новая системная шина ISA (Industry Standard Architecture), полностью реализующая возможности упомянутого микропроцессора. Количество адресных линий было увеличено на четыре, а данных - на восемь. Таким образом, можно было передавать параллельно уже 16 разрядов данных, а благодаря 24 адресным линиям напрямую обращаться к 16 Мбайтам системной памяти. Количество линий аппаратных прерываний в этой шине было увеличено с 7 до 15, а каналов DMA - с 4 до 7.

Надо отметить, что новая системная шина ISA полностью включала в себя возможности старой 8-разрядной шины, то есть все устройства, используемые в PC XT, могли без проблем применяться и в PC AT 286. Системные платы с шиной ISA позволили выполнять синхронизацию работы самой шины и микропроцессора разными тактовыми частотами, за счет чего устройства, выполненные на платах расширения, могли работать медленнее, чем базовый микропроцессор. Это стало особенно актуальным, когда тактовая частота процессоров превысила 10-12 МГц. Теперь системная шина ISA работает асинхронно на частоте 8 МГц; таким образом, теоретически максимальная скорость передачи может достигать 16 Мбайт/с.

Шина ISA имеет следующие параметры:

а) для IBM PC XT:

- 20 адресных линий (A0 - A19),
- 8 линий данных (двунаправленных),
- максимальная пропускная способность 1.2 Мбайт/сек,
- 6 линий запроса прерывания (IRQ2 - IRQ7),
- 3 линии DMA,
- рабочая частота шины 4.77 МГц.

б) для IBM PC AT:

- 16 линий данных,

- максимально адресуемая память - до 16 Мбайт (224),
- добавлены дополнительные 5 линий IRQ (тактируемые по фронту),
- частичная поддержка множества мастеров шины путем введения дополнительных сигналов,
- пропускная способность 5.3 Мбайт/сек,
- рабочая частота шины 8 МГц.

С появлением новых микропроцессоров, таких как i80386 и i486, стало очевидно, что одним из вполне преодолимых препятствий на пути повышения производительности компьютеров с этими микропроцессорами является системная шина типа ISA. Дело в том, что возможности этой шины для построения высокопроизводительных систем следующего поколения были практически исчерпаны.

Шина EISA обеспечивает больший возможный объем адресуемой памяти, 32-разрядную передачу данных, в том числе и в режиме DMA, улучшенную систему прерываний и арбитраж DMA, автоматическую конфигурацию системы и плат расширения.

Шина EISA (Extended Industry Standard Architecture) первоначально была ориентирована на вполне конкретную область применения - на компьютеры, оснащенные высокоскоростными подсистемами внешней памяти на жестких магнитных дисках с буферной кэш-памятью. Такие компьютеры до сих пор используются в основном в качестве мощных файл-серверов или рабочих станций.

В EISA-разъем на системной плате компьютера, помимо, разумеется, специальных EISA-плат, может вставляться либо 8-, либо 16-разрядная плата расширения, предназначенная для обыкновенной PC AT с шиной ISA. Это обеспечивается поистине гениальным, но простым конструктивным решением. EISA-разъемы имеют два ряда контактов, один из которых (верхний) использует сигналы шины ISA, а второй (нижний) - соответственно EISA.

Контакты в соединителях EISA расположены так, что рядом с каждым сигнальным контактом находится контакт "земля". Благодаря этому сводится к минимуму вероятность генерации электромагнитных помех, а также уменьшается восприимчивость к та-

ким помехам. Шина EISA позволяет адресовать 4-Гбайтное адресное пространство, доступное микропроцессорам i80386/486.

Однако доступ к этому пространству могут иметь не только центральный процессор, но и платы управляющих устройств типа bus master - главного абонента (то есть устройства, способные управлять передачей данных по шине), а также устройства, организующие режим DMA.

Стандарт EISA поддерживает многопроцессорную архитектуру для "интеллектуальных" устройств (плат), оснащенных собственными микропроцессорами. Поэтому данные, например, от контроллеров жестких дисков, графических контроллеров и контроллеров сети могут обрабатываться независимо, не загружая при этом основной процессор. Теоретически максимальная скорость передачи по шине в так называемом пакетном режиме (burst mode) может достигать 33 Мбайт/с. В обычном (стандартном) режиме скорость передачи по шине EISA не превосходит, разумеется, известных значений для ISA.

На шине EISA предусматривается метод централизованного управления, организованный через специальное устройство - системный арбитр. Таким образом поддерживается использование ведущих устройств на шине, однако предусматривается также предоставление шины запрашивающим устройствам по циклическому принципу.

Как и для шины ISA, в системе EISA имеется 7 каналов DMA. Выполнение DMA-функций полностью совместимо с аналогичными операциями на ISA- шине, хотя они могут происходить и несколько быстрее. Контроллеры DMA имеют возможность поддерживать 8-, 16- и 32-разрядные режимы передачи данных. В общем случае возможно выполнение одного из четырех циклов обмена между устройством DMA и памятью системы. Это - ISA - совместимые циклы, использующие для передачи данных 8 тактов шины; циклы типа А, исполняемые за 6 тактов шины; циклы типа В, исполняемые за 4 такта шины, и циклы типа С (или burst), в которых передача данных происходит за один такт шины. Типы циклов А, В и С поддерживаются 8-, 16- и 32-разрядными устройствами, причем возможно автоматическое изменение размера (ширины) данных при передаче в не соответствующую размеру память.

Большинство ISA-совместимых устройств, использующих DMA, могут работать почти в 2 раза быстрее, если они будут запрограммированы на применение циклов А или В, а не стандартных (и сравнительно медленных) ISA-циклов. Такая производительность достигается только путем улучшения арбитража шины, а не в ущерб совместимости с ISA.

Приоритеты DMA в системе могут быть либо "вращающимися" (переменными), либо жестко установленными. Линии прерывания шины ISA, по которым запросы прерывания передаются в виде перепадов уровней напряжения (фронтов сигналов), сильно подвержены импульсным помехам. Поэтому в дополнение к привычным сигналам прерываний на шине ISA, активным только по своему фронту, в системе EISA предусмотрены также сигналы прерываний, активные по уровню. Причем для каждого прерывания выбор той или иной схемы активности может быть запрограммирован заранее. Собственно прерывания, активные по фронту, сохранены в EISA только для совместимости со "старыми" адаптерами ISA, обслуживание запросов на прерывание которых производит схема, чувствительная к фронту сигнала.

Понятно, что прерывания, активные по уровню, менее подвержены шумам и помехам, нежели обычные. К тому же (теоретически) по одной и той же физической линии можно передавать бесконечно большое число уровней прерывания. Таким образом, одна линия прерывания может использоваться для нескольких запросов.

Для компьютеров с шиной EISA предусмотрено автоматическое конфигурирование системы. Каждый изготовитель плат расширения для компьютеров с шиной EISA предоставляет вместе с этими платами и специальные файлы конфигурации. Информация из этих файлов используется на этапе подготовки системы к работе, которая заключается в разделении ресурсов компьютера между отдельными платами.

Для "старых" плат адаптеров пользователь должен сам подобрать правильное положение DIP-переключателей и перемычек, однако сервисная программа на EISA-компьютерах позволяет отображать установленные положения соответствующих переключателей на экране монитора и дает некоторые рекомендации по пра-

вильной их установке. Помимо этого, в архитектуре EISA предусматривается выделение определенных групп адресов ввода - вывода для конкретных слотов шины - каждому разъему расширения отводится адресный диапазон 4Кбайта. Это также позволяет избежать конфликтов между отдельными платами EISA. Кроме того, шина по-прежнему тактируется частотой около 8 МГц, а скорость передачи увеличивается в основном благодаря увеличению разрядности шины данных. Отметим, что шина EISA имеет следующие параметры:

- 32 - разрядный режим передачи
- максимальная пропускная способность - до 33 Мбайт/сек
- 32 - разрядная адресация памяти, что обеспечивает до 4 Гбайт адресуемого пространства памяти
- множество мастеров шин
- программируемые прерывания по уровню или по фронту синхросигнала
- автоматическая конфигурация плат

Шина Nubus обладает примерно теми же характеристиками, что и ISA.

Шина Multibus-II была разработана в 1985 г. как развитие широко применяемого в промышленной автоматике стандарта Multibus. Multibus-II является 32-разрядной шиной и может работать со скоростью управляющего процессора - вплоть до достижения пропускной способности 80 Мбайт/с. В отличие от других рассматриваемых здесь шин, Multibus обладает возможностью высокоскоростной передачи сообщений между различными управляющими устройствами. При этом механизм передачи позволяет организовывать "интеллектуальное" взаимодействие между процессорами и контроллерами. Это особенно важно при создании многопроцессорных систем и построении сложных комплексов промышленной электроники.

Шина содержит пять магистралей (рис. 76), логическая организация которых приспособлена для решения задач определенного класса:

- i PSB - магистраль параллельная системная межмашинная;

- i LBX - магистраль параллельная локального расширения;
- i SSB - магистраль последовательная системная;
- i SBX - магистраль параллельная расширения ввода - вывода;
- MDMA - параллельная магистраль каналов прямого доступа к памяти.

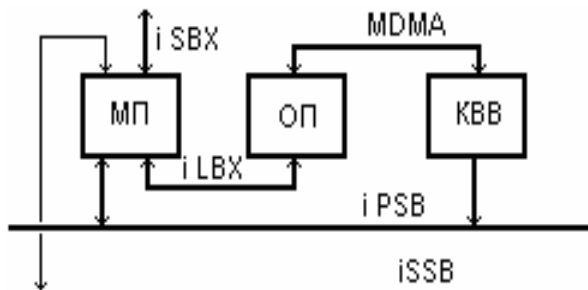


Рис. 76

Одновременное использование магистралей i PSB, i LBX, MDMA обеспечивает суммарную пропускную способность 96 Мбайт/с без учета скорости передачи по магистрали i SBX, предназначенной специально для организации ввода - вывода. Параллельные магистрали используют синхронное стробирование, повышающее помехоустойчивость. Магистрали i PSB и i LBX содержат контрольные разряды по четности, а магистраль i SSB использует 16 - разрядный код циклического избыточного кодирования.

Основная системная магистраль i PSB объединяет все модули системы и через нее осуществляется управление и контроль за ее работой. В ее состав входят 32 совмещенные линии адреса/данных, пять линий параллельного арбитража, с помощью которых осуществляется процедура приоритетов при прерываниях.

Шина MC (MicroChannel) появилась в 1987 г. в компьютерах PS/2. Достаточно быстрая (до 20 МГц, до 76 Мбайт/с) и широкая (32 бита), шина содержала ряд удачных архитектурных решений и вполне могла бы бороться за лидерство среди системных шин. Она обладает следующими особенностями:

- 8/16/32 - разрядные линии передачи данных,
- прерывания по уровню сигнала (в отличие от ISA, где прерывания - по фронту синхросигнала),
- 24 или 32 адресных линии (адресация до 4 Гбайт памяти),
- автоматическая конфигурация плат (на основе информации в ROM этих плат),
- асинхронный протокол передачи данных.

Шина Sbus разработана в 1989 г. для работы с частотой до 25 МГц. Она предназначена для передачи 32-разрядных данных. Ее особенностью являются возможность автоматически транслировать виртуальные адреса в физические, распознавать ошибки при передаче данных и инициировать повторы.

Шина Mbus создана в 1990 г. и предназначена для передачи 64-разрядных данных. Mbus допускает совместное использование с другими шинами, имеет портативные варианты исполнения и предусматривает возможности передачи сообщений.

Шина SCSI (Small Computer System Interface) регламентирован стандартом IEC 9316, который унифицирует основные уровни для базовых типов периферийных устройств, главным образом накопителей магнитных дисков, АЦПУ, а также возможности расширения функций посредством специальных кодов и полей. В интерфейсе используется логическая адресация всех блоков данных и возможность считывания с устройств прямого доступа информации о числе имеющихся блоков.

Максимальная скорость передачи данных составляет до 4 Мбайт/сек, длина кабеля до 6 м при использовании обычных приемопередатчиков и до 25 м дифференциальных приемопередатчиков. Архитектура интерфейса предусматривает несколько видов организации взаимодействия задатчиков (инициаторов) и исполнителей (приемников) с использованием необязательного распределенного арбитража. Время арбитража не превышает 10 мкс.

Дополнительные возможности такие: два варианта физической реализации, использование четности, синхронная передача данных и др. Команды разделены на обязательные, расширенные, необязательные и уникальные. Устройства выполняют все обязательные команды для данного типа устройств команды, а также

ряд других команд. Кроме того, в стандарте определены расширенные команды для устройств прямого доступа, постоянные команды для всех типов устройств, уникальные команды для жестких дисков, ленточных накопителей, принтеров, оптических дисков, процессоров, байты состояния всех типов устройств.

Максимальное число подключенных устройств - 8. Каждое устройство идентифицируется соответствующим разрядом, размещаемым на линии данных. SCSI-2 является одной из "старых" периферийных шин, используемых, с доработками, и поныне. Спецификация SCSI разрабатывалась американским институтом национальных стандартов ANSI. Чуть позже она расширилась до SCSI-2 и SCSI-3.

Типичная SCSI обладает следующими характеристиками:

- 8 - разрядная параллельная шина ввода-вывода,
- каждый адаптер может поддерживать до 7 устройств,
- поддерживаются различные устройства (CD-ROM, ленточные накопители, сканеры, магнитооптические устройства и т. д.),
- пропускная способность 4 Мбайт/сек,
- поддержка синхронной и асинхронной схем передачи данных.

SCSI-2 расширяет возможности основного стандарта. Она имеет максимальную пропускную способность до 10 Мбайт/сек при 8 - разрядной шине и до 40 Мбайт/сек - при 32-разрядной шине. Существует несколько спецификаций приложений для SCSI:

- Narrow SCSI 8-разрядная версия SCSI,
- Wide SCSI 16- и 32-разрядные версии SCSI-2,
- Fast SCSI SCSI-2, которая поддерживает скорость передачи до 10 Мбайт/сек

Разработчики компьютеров, системные платы которых основывались на микропроцессорах i80386/486, стали использовать отдельные шины для памяти и устройств ввода-вывода. Это позволило максимально задействовать возможности оперативной памяти, так как именно в этом случае память может работать с наивысшей для нее скоростью. Тем не менее при таком подходе вся система не может обеспечить достаточной производительности, так как устройства, подключенные через разъемы расшире-

ния, не могут достичь скорости обмена, сравнимой с процессором. В основном это касается работы с контроллерами накопителей и видеоадаптерами.

Для решения данной проблемы стали использовать так называемые локальные (local или mezzanine) шины, которые непосредственно связывают процессор с контроллерами периферийных устройств. Известны две стандартные локальные шины: **VL-bus** (или VLB), предложенная ассоциацией VESA (Video Electronics Standards Association), и **PCI** (Peripheral Component Interconnect), разработанная фирмой Intel. Обе эти шины, предназначенные, вообще говоря, для одного и того же - для увеличения быстродействия компьютера, позволяют таким периферийным устройствам, как видеоадаптеры и контроллеры накопителей, работать с тактовой частотой 33 МГц и выше. Обе эти шины используют разъемы типа MCA.

Шина VL-Bus является расширением шины процессора 486. Выводы процессора подключаются непосредственно к контактам разъема шины. В некоторых платах адаптеров VL-Bus имеются буферы для хранения данных на время ожидания готовности периферийного устройства. Таким образом, схемная реализация VL-bus оказывается более, дешевой и простой, чем, например, PCI. Спецификация VESA, в частности, предусматривает, что к шине, которая является локальной 32-разрядной шиной системного микропроцессора, может подключаться до трех периферийных устройств. В качестве таких устройств в настоящее время выступают контроллеры накопителей, видеоадаптеры и сетевые платы.

Конструктивно VL-bus выглядит как короткий соединитель типа MCA (112 контактов), установленный, например, рядом с разъемами расширения ISA или EISA. При этом 32 линии используются для передачи данных и 30 - для передачи адреса. Максимальная скорость передачи по шине VL-bus теоретически может составлять около 130 Мбайт/с.

Заметим, что в настоящее время шина VL-bus представляет из себя сравнительно недорогое дополнение для компьютеров с шиной ISA, причем с обеспечением обратной совместимости. Появилась версия 2.0 шинной архитектуры VL-Bus, в которую введены такие новшества, как мультиплексированный 64-

разрядный канал данных, буферизация сигналов для работы с быстродействующими системными платами и более высокая максимальная тактовая частота - 50 МГц. Количество разъемов расширения увеличится до трех разъемов на 40 МГц и до двух на 50 МГц. Ожидаемая скорость передачи теоретически должна возрасти до 400 Мбайт/с.

Стандарт IEEE 896.1-1988, названный Futurebus+, претендует на роль шины завтрашнего дня для систем массового применения. Стандарт Futurebus+ был разработан ассоциацией VITA (VFEA International Trade Association) в 1988 г. специально для высокоскоростных систем передачи информации. Требования к Futurebus+ были составлены таким образом, чтобы преодолеть все ограничения, присущие VME в телекоммуникационных системах. Ширина Futurebus+ - до 256 бит, максимальная скорость - 3,2 Гбайт/с, рабочая частота ограничивается лишь возможностями управляющего процессора.

Для сложных высокоскоростных шин, помимо упомянутых выше "мостов", применяются так называемые mezzanine-bus - более простые и "узкие" шины, сопрягаемые с основной без использования дополнительной управляющей электроники. Для Futurebus+ такими mezzanine-bus являются Sbus и PCI.

Шина PCI обладает несколькими преимуществами перед основной версией VL-Bus. В соответствии со спецификацией PCI к шине могут подключаться до 10 устройств. Это, однако, не означает использования такого же числа разъемов расширения - ограничение относится к общему числу компонентов, в том числе расположенных и на системной плате. Поскольку каждая плата расширения PCI может разделяться между двумя периферийными устройствами, то уменьшается общее число устанавливаемых разъемов.

Шина PCI может использовать 124-контактный разъем (32-разрядная) или 188-контактный разъем (64-разрядная передача данных), при этом теоретически возможная скорость обмена составляет соответственно 132 и 264 Мбайт/с. На системных платах устанавливаются обычно не более трех разъемов.

Предполагается, что стандарт PCI лучше соответствует растущим потребностям в скоростной обработке данных на настоль-

ных машинах, поскольку превосходит стандарт VL-Bus по сложности, гибкости и функциональной насыщенности. Windows принесла в мир ПК полноцветную графику. Процессор 486 выполняет пересылки данных по 32-разрядной шине, тактируемой частотой 33 МГц. Как только выдаваемый им мощный поток графических данных попадает на шину ISA, он упирается в "узкое горло". Эта шина работает на частоте всего лишь 8 МГц, а ее разрядность равна 16. По мере того как в прикладных программах начинают все шире использоваться многоцветная графика, "живое" видео и рендеринг трехмерных изображений, разработчикам систем и периферийных устройств пришлось предусмотреть другой способ связи с узлами машины, требующими наиболее интенсивного обмена данными.

Стандартная локальная шина обеспечивает единообразный способ подключения устройств к быстродействующей шине процессора и тем самым позволяет устранить "узкие места" во всех новых ПК. Шина PCI поддерживает 32-разрядный канал передачи данных между процессором и периферийными устройствами, работает на высокой тактовой частоте (33 МГц) и имеет максимальную пропускную способность 120 Мбайт/с. Кроме того, шина PCI в некоторой степени обеспечивает обратную совместимость с существующими периферийными устройствами, рассчитанными на шину ISA.

В стандарте PCI предусмотрены контроллер и акселератор, образующие локальную шину, не связанную с шиной процессора. В ней используется несколько способов повышения пропускной способности. Один из них - блочная передача последовательных данных. Если данные не являются последовательными, требуется дополнительное время на установку адреса каждого их элемента. Шина PCI создает между ЦП и периферийными устройствами некоторый промежуточный уровень. В результате получается процессорно-независимая шина, как ее называет Intel. Ее легко подключить к самым различным процессорам, в их числе Pentium (Intel), Alpha (DEC), MIPS R4400 и PowerPC (Motorola, Apple и IBM).

Для производителей систем это означает снижение затрат на разработку, так как с процессорами разного типа можно ис-

пользовать одни и те же элементы и устройства. Стандарт PCI предусматривает обширный список дополнительных функций. К ним относится автоматическая конфигурация периферийных устройств, позволяющая пользователю устанавливать новые устройства без особых проблем.

PCI поддерживает целый спектр периферийных устройств и обладает средствами управления передачей данных (что освобождает процессор от рутинной возни с трафиком). Нет нужды говорить, что все обмены по шине буферизованы. PCI легко совместима с большинством известных шин. Разработаны и реализованы в виде стандартных микросхем многочисленные "мосты"; PCI/ISA, PCI/EISA, PPC/PCI и другие. Многие производители ПК практикуют также слоты двойного назначения - например, PCI/ISA, позволяющие на одно и то же место устанавливать устройства ввода-вывода в различных стандартах.

Интерфейс МПИ с мультиплексированными линиями адреса и данных предназначен для обеспечения информационной и электрической совместимости устройств системы. Он реализуется на основе магистрали и логических узлов, входящих в каждое подключаемое к ней устройство. Устройства в совокупности составляют единое адресное пространство магистрали.

В интерфейсе коды адреса и данных передаются по одной и той же группе сигнальных линий мультиплексированной шине обмена информацией) с разделением во времени. Принцип работы интерфейса при передаче данных — асинхронный, а при передаче адреса — синхронный.

В каждый момент времени на магистрали может выполняться один из трех видов взаимодействий подключенных к ней устройств: передача управления магистралью, адресный обмен (одиночный или блочный), прерывание.

Передача управления магистралью осуществляется в соответствии со схемой приоритета. Приоритет устройства определяется его положением на линии «разрешение на захват магистрали» (РЗМ) относительно других устройств. Приоритет устройства убывает по мере удаления устройства от микропроцессора, управляющего захватом магистрали, в направлении распространения сигнала РЗМ. При процедуре передачи управления магистралью

активное устройство, готовое к выполнению функции ведущего, асинхронно выставляет запрос на захват магистрали. МП выдает разрешение на захват магистрали после завершения текущего цикла обмена информацией или другого взаимодействия.

Адресный обмен строится по принципу ведущий — ведомый. В любой момент времени на магистрали взаимодействуют только одно ведущее и одно ведомое устройство. Ведущее устройство инициирует обмен информацией и задает его режим. При этом интерфейс может обеспечить режимы одиночного (обязательного) и блочного (необязательного) обменов данными.

Прерывание выполняемой программы МП осуществляет по запросам ВУ. При обработке запроса на прерывание процессор запоминает состояние прерванной программы и продолжает ее после завершения прерывающей программы. Контроллер ВУ, запросившего прерывание, по разрешению процессора выдает вектор прерывания, определяющий вход в процедуру обработки программы данного прерывания. Разрешение на выдачу вектора прерывания МП выдает в соответствии с многоуровневой системой приоритетов.

В зависимости от формата адреса процессора и диспетчера памяти пространство магистрали может составлять 64, 128, 256, 512, 1024, 2048, 4996, 8192 или 16 384 Кбайт. Во всех случаях 8 Кбайт адресного пространства магистрали используются для адресации ВУ, остальной для ячеек внутренних запоминающих устройств.

По магистрали информация передается в двоичном позиционном коде. Длина слова данных составляет 8 или 16 бит, а формат передаваемого адреса - 16 - 24 бит.

Интерфейс Unibus содержит магистраль из 56 сигнальных линий. Все устройства подсоединяются к этим линиям параллельно. Пять симплексных сигнальных линий используются для управления шиной приоритета, остальные 51 линий являются дуплексными; 18 адресных линий используются ведущим устройством для выборки ведомого устройства, с которым предстоит установить связь. Одна из линий адреса задает байт, к которому при операциях с байтами происходит обращение; 16 линий данных используются для передачи информации между ведущим и ведомым

устройствами. Две линии управления задают одну из четырех возможных операций обмена (два режима ввода и два - вывода).

Все передачи по общей шине осуществляются по методу «запрос - ответ». Такая организация взаимодействия позволяет объединить на магистрали устройства различного быстродействия. Для взаимной синхронизации ведущего и ведомого устройств используются две линии синхронизации. Для передачи управления магистралью ведущему устройству используется 11 линий приоритета (линии запроса, разрешения и подтверждения выбора). Для осуществления ввода-вывода данных без участия программы предусмотрен режим прямого доступа к памяти.

Интерфейс И-41 является одним из вариантов интерфейса Multibus, объединяющего стандартизованные интерфейсы IEEE, VME - bus, AMS - bus и др., с сохранением состава линий и их функций.

Интерфейсы 2 уровня обеспечивают объединение внешних устройств и устройств связи с объектами (УСО), которые используются в тех случаях, когда ВУ и УСО не имеют встроенного системного интерфейса и не могут подключаться непосредственно к системной магистрали. Наибольшее распространение здесь получили интерфейс ИРПС для радиального подключения устройств с последовательной передачей информации и интерфейс ИРПР для подключения устройств с параллельной передачей информации. С их помощью подключаются практически все периферийные устройства (дисплеи, принтеры, клавиатура, графопостроители и т. д.), за исключением внешних запоминающих устройств, предъявляющих более высокие требования к пропускной способности интерфейса.

В качестве интерфейса УСО могут быть использованы магистраль КАМАК или специальные интерфейсные платы - контроллеры, обеспечивающие подключение модулей УСО к системному интерфейсу. Сопряжение малого интерфейса с системной магистралью осуществляется при помощи контроллера К (рис. 3.18).

Интерфейсы 3 уровня предназначены для объединения датчиков и исполнительных устройств. Большое разнообразие датчиков и исполнительных устройств на сегодняшний день привело к разработке огромного числа этих интерфейсов. Интерфейсы

4 уровня представляют собой интерфейсы устройств передачи данных (УПД). К ним относятся интерфейсы телеграфных, телефонных, высокочастотных, оптоволоконных и других каналов для передачи данных на большие расстояния. Сюда же относятся интерфейсы распределенных систем управления общего и специального назначения (КАМАК МЭК - 640, МЭК - 625 - 1 последовательный, ИЛПС - 2 и др.) и интерфейсы локальных сетей общего назначения (Р - 802 и др.)

Интерфейсы 5 уровня включают внешние относительно микропроцессорной системы интерфейсы. Соединение внешнего интерфейса с системным осуществляется при помощи специального адаптера интерфейсов.

В большинстве из рассмотренных интерфейсов применяют три режима передачи данных (и соответственно три типа каналов связи): симплексный, полудуплексный и дуплексный.

Симплексный режим обеспечивает одностороннюю связь между передатчиком и приемником, территориально разнесенных между собой.

Полудуплексный режим обеспечивает двусторонний обмен данными между двумя точками, в каждой из которых имеется передатчик и приемник, но одновременная передача в двух направлениях невозможна. Для изменения направления передачи требуется некоторое время переключения (коммутации).

При передаче больших объемов информации применяются дуплексный режим, обеспечивающий одновременную передачу информации в обоих направлениях.

Обмен информацией в интерфейсах может производиться с использованием синхронного (обмен со стробированием) и асинхронного (обмен с квитированием) принципов обмена. В первом случае устройство - источник (контроллер) определяет темп выдачи и приема информации и синхронизирует все процессы, связанные с трансляцией данных. Обычно синхронизируется прохождение в линии каждого бита, группы битов (символа) и сообщения.

Асинхронный принцип передачи в интерфейсах, как правило, основан на режиме запроса - ответа. В этом случае устройство - источник по одной из линии интерфейса вырабатывает сигнал о выдаче данных на ШД и направляет его в устройство - приемник.

Приемник фиксирует поступление сигнала готовности источника, принимает данные и извещает об этом источник сигналом, появляющимся на другой линии (строб готовности приемника). Источник, восприняв ответ, снимает передаваемые данные. Таким образом, интервал времени, в течение которого источник выводит данные на шину интерфейса, является переменным и зависит от характеристик как самого источника, так и приемника сигналов, а также характеристик линий связи.

Хотя при синхронной передаче данных по сравнению с асинхронной более эффективно используется канал связи и достигается лучшая помехозащищенность передаваемых данных, в интерфейсах автоматизированных систем научных исследований применяют, как правило, асинхронный способ передачи. Это обусловлено возможностью передавать в асинхронном режиме данные со скоростью, соответствующей быстродействию того устройства, с которым в данный момент времени происходит обмен информацией (автоматическая подстройка скорости передачи данных).

Интерфейс AGP предназначен для вывода информации на внешние устройства, в том числе отображения данных. Она содержит шину и устройство передачи информации (видеоускоритель), образующие интерфейсную схему. В настоящее время наибольшее применение получил интерфейс AGP

В начале 1997 г. фирмой Intel был разработан новый стандарт для вывода графики, получивший название AGP (Accelerated Graphics Port). Здесь видеопамять располагается не на графическом адаптере, а в ОЗУ компьютера. В процессе обработки информации процессор автоматически выделяет необходимый объем памяти для вывода графики. Физически это будет реализовано в виде добавки для шины PCI и полностью прозрачно для нее.

AGP работает на частоте основной памяти (66 МГц) и в обычном режиме (x1), при котором данные передаются только по переднему фронту тактового сигнала, дает возможность достичь пиковой пропускной способности 266 Мбайт/с, а в режиме (x2), при котором данные передаются и по переднему, и по заднему фронту тактового сигнала, при этом пропускная способность достигает значения в 532 Мбайт/с.

AGP работает в двух режимах. Первый из них основан на традиционной модели DMA, а второй - на новой модели DIME. В зависимости от выбранного режима данные по-разному распределяются между основной и локальной памятью, что, в свою очередь, влияет на качество отображаемой картинки и частоту смены кадров.

В режиме DMA для графики используется только локальная память видеоускорителя, а данные, расположенные вне ее, предварительно загружаются в локальную память и лишь затем обрабатываются видеопроцессором. При этом AGP выполняет роль быстрой шины.

В режиме DIME для построения изображения видеоускоритель использует локальную и системную память. При этом любая структура данных может располагаться как в локальной, так и в системной памяти. Данные не копируются предварительно из системной памяти в локальную, а интерпретируются «на месте».

Необходимо отметить одну важную особенность AGP, которая состоит в том, что память под текстуры выделяется операционной системой по требованию исполняемой программы и остается доступной для него. Поскольку текстура может занимать более одной страницы оперативной памяти (более 4 Кбайт), то в спецификацию AGP включена таблица переадресации графики, содержимое которой должно быть согласовано с таблицами переадресации операционной системы. Таким образом, поддержка AGP не может ограничиваться драйвером производителя, здесь требуется поддержка на уровне операционной системы. Такая поддержка реализована, например, в версии Windows 98.

Преимуществом при использовании AGP является следующее:

- интерфейс реализован по принципу «соединение точка-точка», при котором отсутствуют проблемы с арбитражем шин,
- в интерфейсе предусмотрены отдельные шины для передачи команд и данных, повышающие пропускную способность интерфейса,
- применение конвейеризации и технологии отложенного выполнения команд, позволяющие вплотную приблизиться к теоретическому скоростному пределу интерфейса.

Принимая во внимание перечисленные преимущества интерфейса AGP разработано большое количество видеоускорителей, совместимых со спецификацией AGP, среди которых можно отметить Asus 3DexPlover 3000, ATI Xpert@Work, Diamond Viper330, Matrox Millenium II, STB Velocity 128 и др.

В заключение необходимо отметить, что единого подхода по оценке быстродействия пока нет. Связано это с тем, что, во-первых, разные видеоускорители на аппаратном уровне реализуют очень разные наборы функций, а, во-вторых, различные видеопрограммы используют существенно разные функции. Большое разнообразие функций является единственной главной причиной, не позволяющей однозначно определить быстродействие. Однако проведенные приближенные оценки показывают, что для большинства программ трехмерной графики быстродействие микропроцессорных систем вывода видеoinформации с использованием видеоускорителей, использующих шину PCI, и систем с использованием интерфейса AGP приблизительно одинаково.

12. Организация памяти МПС

Память современных вычислительных систем имеет иерархическую многоуровневую структуру. Чем выше уровень, тем выше требуемое быстродействие соответствующей памяти. Основная или оперативная память относится к верхнему уровню памяти.

Сравнительно небольшая емкость ОП (до нескольких десятков Мбайт) компенсируется практически неограниченной емкостью внешних запоминающих устройств на магнитных дисках (до нескольких Гбайт). Однако эти устройства сравнительно медленные, и время обращения за данными для дисков составляет десятки миллисекунд. Поэтому вычислительный процесс должен протекать с возможно меньшим числом обращений к внешним запоминающим устройствам и максимально возможным использованием ОП.

Быстродействие ОП часто оказывается недостаточным для обеспечения требований, предъявляемых к скорости работы МПС. Это проявляется в несоответствии пропускных способностей про-

цессора и ОП. Возникающая проблема выравнивания их пропускных способностей решается путем использования буферных памяти небольшой емкости и повышенного быстродействия. Они используются для хранения команд и данных, относящихся к обрабатываемому участку программы.

Оперативная память является наиболее дефицитным ресурсом в вычислительных системах, которым надо пользоваться экономно и эффективно. Проблема усложняется при переходе к мультипрограммным системам, так как в них ОП одновременно использует несколько программ (заданий). В таких системах важным стоит вопрос исключения несанкционированного воздействия одних программ на другие. Это достигается с помощью механизма защиты памяти.

Защита памяти. Если в памяти одновременно могут находиться несколько независимых программ, необходимы специальные меры по предотвращению или ограничению обращений одной программы к областям памяти, используемым другими программами. Программы могут содержать такие ошибки, которые, если этому не воспрепятствовать, приводят к искажению информации, принадлежащей другим программам. Последствия таких ошибок особенно опасны, если разрушению подвергнутся программы операционной системы. Другими словами, надо исключить воздействие программы пользователя на работу программ других пользователей и программ операционной системы.

Чтобы воспрепятствовать разрушению одних программ другими, достаточно защитить область памяти данной программы от попыток записи в нее со стороны других программ, а в некоторых случаях и своей программы (защита от записи), при этом допускается обращение других программ к этой области памяти для считывания данных.

В других случаях, например при ограничениях на доступ к информации, хранящейся в системе, необходимо иметь возможность запрещать другим программам производить как запись, так и считывание в данной области памяти. Такая защита от записи и считывания помогает отладке программы, при этом осуществляется контроль каждого случая выхода за область памяти своей программы.

Для облегчения отладки программ желательно выявлять и такие характерные ошибки в программах, как попытки использования данных вместо команд или команд вместо данных в собственной программе, хотя эти ошибки могут и не разрушать информацию.

Необходимо отметить следующие варианты дифференцированной защиты при различных операциях с памятью:

— задается отношение к области памяти чужой программы, определяющее, относится защита памяти только к операции записи или к любому обращению в память;

— задается одно из следующих отношений к области памяти собственной программы;

а) разрешается доступ к данному блоку как для записи, так и для считывания;

в) разрешается обращение любого вида, но по адресу, взятому только из счетчика команд;

г) разрешается обращение по адресу из любого регистра, кроме счетчика команд.

Если нарушается защита памяти, исполнение программы приостанавливается и вырабатывается запрос прерывания по нарушению защиты памяти.

Защита от вторжения программ в чужие области памяти может быть организована различным образом, при этом реализация защиты не должна заметно снижать производительность МПС и требовать слишком больших аппаратных затрат.

Защита отдельных ячеек памяти. В управляющих вычислительных комплексах необходимо обеспечить возможность отладки новых программ параллельно с функционированием находящихся в памяти рабочих программ, управляющих технологическим процессом. Это может быть достигнуто выделением в каждой ячейке памяти специального «разряда защиты». Установка «1» в этот разряд запрещает производить запись в данную ячейку.

В системах с мультипрограммной обработкой большого числа программ защищаются не отдельные ячейки, а области памяти или блоки, на которые делится память, при этом часто предусматривается возможность указывать для разных программ различные

допустимые режимы обращения к отдельным областям или блокам памяти.

Существует несколько методов защиты памяти, среди которых можно выделить метод граничных регистров и метод ключей защиты.

Метод граничных регистров состоит во введении двух граничных регистров, указывающих верхнюю и нижнюю границы области памяти, куда программа имеет право доступа. При каждом обращении к памяти проверяется, находится ли используемый адрес в установленных границах; при выходе за границы обращение к памяти подавляется и формируется запрос прерывания, передающий управление операционной системе.

Содержание граничных регистров устанавливаются операционной системой перед тем, как для очередной исполняемой программы начнется активный цикл. Если для динамического распределения памяти используется базовый регистр, то он одновременно определяет и нижнюю границу. Верхняя граница подсчитывается операционной системой в соответствии с длиной программы оперативной памяти.

Метод ключей защиты. По сравнению с предыдущим данный метод является более гибким: он позволяет организовать доступ программы к областям памяти, расположенным не подряд.

Память в логическом отношении делится на одинаковые блоки. Каждому блоку памяти ставится в соответствие код, называемый ключом защиты памяти, а каждой программе, принимающей участие в мультипрограммной обработке, присваивается код ключа программы. Доступ программы к данному блоку памяти для чтения и записи разрешен, если ключи совпадают или один из них имеет код «0». Коды ключей защиты памяти хранятся в специальной памяти защиты ключей, более быстродействующей, чем ОП.

Пропускная способность процессора и МПС. Кэш – память. Непрерывный рост производительности (скорости работы) МПС проявляется, в первую очередь, в повышении скорости работы электронных схем, а также специальных архитектурных решений — конвейерная и векторная обработка данных, кэширование памяти и др.

Быстродействие оперативной памяти также растет, но все время отстает от быстродействия аппаратурных средств процессора, в значительной степени потому, что одновременно происходит опережающий рост ее емкости. Это делает более трудным уменьшение времени цикла работы памяти.

Без согласования пропускных способностей процессора и памяти невозможно в системе реализовать производительность, соответствующую быстродействию процессора. Преодолеть указанное противоречие и согласовать пропускные способности памяти и процессора помогают специальные структурные решения.

Конвейеризация процедур цикла выполнения команды (рабочего цикла машины) в простейшем случае предполагает выполнение параллельно во времени операции в АЛУ с выборкой из памяти следующей команды.

Буферизация — использование включенных между процессором и ОП существенно более, чем ОП, быстродействующих буферных памяти сравнительно небольшой емкости. На рис. 76 показана структура процессора, содержащая буфер команд и буфер операндов.

Представленные на рис. 76 схемы буферной памяти скрыты от программиста в том смысле, что он не может их адресовать, может даже не знать об их существовании. Поэтому они получили название кэш - памяти. Структура высокопроизводительных 32- и 64-разрядных МП содержит объединенную кэш - память для фрагментов программ и групп данных, при этом в ряде случаев наряду с кэш - памятью сохраняется небольшой буфер на несколько команд. Обмен информацией между кэш-памятью и микропроцессором осуществляется высокоскоростной локальной шиной, при этом кэш-команд имеет свою шину, а кэш-данных свою. В последних разработках (Pentium III и IV) работа шин осуществляется независимо друг от друга, за что они получили название USB - двойной независимой шины.

Кэш - память чаще всего располагается на одном кристалле с процессором, но может располагаться и вне кристалла, но при этом она находится на той же плате вычислителя и служит высокоскоростным буфером между процессором и относительно медленной основной памятью. При некоторых обращениях к памяти

соответствующие значения заносятся в кэш, и в ходе следующих операций чтения по тем же адресам обращения происходят только к кэш - памяти.



Рис. 76

Для обращения к кэш - памяти, размещенной вне кристалла процессора, но на одной плате с процессорной БИС, может потребоваться несколько циклов памяти, тогда как при обращении к кэш - памяти, находящейся в процессорном кристалле, может оказаться достаточно одного такого цикла; однако даже размещение кэш - памяти на одной плате с процессором позволяет избежать большого числа циклов ожидания, которые неизбежны при работе с памятью, расположенной на отдельной плате и взаимодействующей с процессором через системную шину, а также снизить нагрузку системной шины.

Кэш - память может использоваться для хранения либо команд, либо данных, либо информации обоих этих видов. Выборка ее содержимого может производиться произвольным образом, и, следовательно, в адресном диапазоне кэш-памяти можно разместить командные циклы с входящими в них командами переходов.

Производительность кэш - памяти определяется временем доступа и вероятностью удачных обращений. Она зависит от объема кэш-памяти и количества битов, записываемых в него при каждом обращении к основной памяти, называемого длиной строки. С увеличением длины строки повышается вероятность того, что следующее обращение будет удачным, т. е. необходимая

информация окажется в кэш-памяти. Если при объеме кэш 4 Кбайт и длине строки 4 байт вероятность удачных обращений составляет 80%, то при увеличении длины строки до 8 байт эта величина может достигнуть 85%. Однако дальнейшее увеличение длины строки не приводит к заметному росту вероятности удачных обращений. Так, например, увеличение длины строки еще в два раза (до 16 байт) позволяет довести вероятность удачных обращения примерно до 87%.

Полная производительность памяти является функцией времени доступа к кэш-памяти, вероятности удачных обращений к кэш и временем обращения к основной памяти, которое происходит при неудачном обращении к кэш. Возможна, например, ситуация, когда в системе памяти вероятность удачных обращений к кэш, время доступа к которому равно 120 нс, составляет 80%, а вероятность обращений к основной памяти с временем доступа 600 нс равна 20%. При этом среднее время доступ к памяти составляет $(0,8 \times 120) + 0,2 \times (600 + 120) = 240$ нс.

При обращении процессора к ОП для считывания в кэш передается блок информации, содержащий нужное слово. При этом происходит опережающая выборка, так как высока вероятность того, что ближайшие обращения будут происходить к словам этого же блока, уже находящимся в кэш-памяти. Это приводит к значительному уменьшению среднего времени, затрачиваемого на выборку данных.

Эффективность кэш-памяти, зависящая от ее емкости, размера слова, соотношения времен считывания слова из кэш-памяти и блока из ОП проявляется в уменьшении среднего времени, затрачиваемого на выборку слова данных.

Можно выделить два типа кэш - памяти:

— с запоминанием новой информации одновременно в кэш и в ОП («сквозное запоминание»), при этом в ОП есть всегда последняя копия хранящейся в кэш информации. Однако в этом случае длинный цикл ОП снижает производительность процессора;

— с запоминанием новой информации только в кэш и копированием ее в ОП только при передаче в другие устройства или при вытеснении из кэш-памяти.

Динамическое распределение памяти. Виртуальная страничная и сегментная память. Эффективное распределение ресурсов памяти между программами не может быть статическим, т. е. не может производиться предварительно до пуска программы. В процессе обработки программ потребности в ресурсах памяти отдельных программ изменяются, что заранее не может быть учтено. Зачастую целесообразнее распределять память между программами динамически непосредственно в ходе вычислительного процесса, т. е. осуществлять динамическое распределение памяти. При этом должна обеспечиваться возможность независимой работы программистов над своими программами, подлежащими мультипрограммной обработке.

Необходимо отметить, что динамическое распределение памяти не должно приводить к дроблению ее свободного пространства — фрагментации памяти, затрудняющему ее использование. Это достигается организацией одноуровневой виртуальной памяти, допускающей адресацию на все адресное пространство. Размер его определяется количеством разрядов, которые могут быть использованы для представления адреса. К тому же в мультипрограммных системах размещение всех исполняемых программ полностью в ОП во многих случаях невыполнимо: программы часто имеют большую длину, а емкости существующих ОП ограничены. Однако нет принципиальной необходимости в том, чтобы вся программа находилась в ОП, так как в любой момент времени работа программы концентрируется на определенных сравнительно небольших участках. Таким образом, в ОП следует хранить только используемые в данный период времени части программ, а неиспользуемые части могут располагаться во ВЗУ. Программируя свою программу, программист не знает, в комбинации с какими программами будет выполняться его программа, какое место в памяти отведет ей операционная система.

При подготовке программ используются условные адреса. Позднее в процессе выполнения программы операционная система выделяет активным частям программы место в памяти и условные адреса переводятся в исполнительные. Эта процедура получила название динамического распределения памяти.

Осуществление динамического распределения чисто программным путем привело бы к значительным потерям машинного времени. Целесообразнее пользоваться для этой цели аппаратными средствами.

Один из способов динамического распределения памяти основан на использовании базовых регистров. Операционная система каждой исполняемой программе ставит в соответствие свой базовый адрес. Базовые адреса программ находятся в общих регистрах. При выполнении программы реальный или физический адрес образуется суммированием базового и относительного адресов. При динамическом распределении памяти с помощью базовых регистров программа (или, по крайней мере, та ее часть, адрес которой преобразуется с помощью одного и того же базового адреса) должна располагаться в последовательных ячейках и вводиться в ОП целиком, хотя в ближайшем цикле активности, возможно, может потребоваться лишь небольшой фрагмент программы.

При рассматриваемом способе динамического распределения памяти свободная память может состоять из несвязанных областей (фрагментация памяти) и для ввода нужной программы может понадобиться сдвиг содержимого памяти. На рис. 77, а показано распределение памяти между программами А, В, С, D, из которых две, например, А и D, являются в данный момент наименее активными и следовательно, могут рассматриваться как кандидаты на удаление во внешнюю память

Если вновь вводимая программа Е (рис. 77, б) больше любой из программ А и D, то для ее размещения в памяти необходимо, как показано на рис. 77, в, сдвигать программы В и С. Это перемещение связано с потерей времени. Более того, в ряде систем подобное перемещение требует выполнения заново операции редактирования связей в программе и новой загрузки программы, что в ряде случаев ведет к существенному снижению быстродействия системы в целом. Эти и некоторые другие недостатки в распределении памяти отсутствуют в виртуальной памяти со страничной организацией.

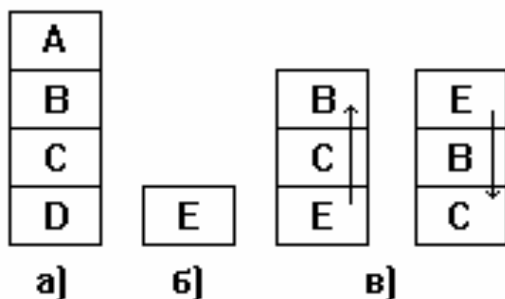


Рис.77

Виртуальная память есть способ, организации памяти мультипрограммной вычислительной системы, при котором достигается гибкое динамическое распределение памяти, устраняется ее фрагментация и создаются значительные удобства для работы программистов. Это удается достигнуть без заметного снижения производительности МП ценой усложнения аппаратуры и операционной системы и процессов их функционирования.

Принцип виртуальной памяти предполагает, что пользователь при подготовке своей программы имеет дело не с физической ОП, действительно работающей в составе вычислительной системы и имеющей некоторую фиксированную емкость, а с виртуальной (т. е. кажущейся) одноуровневой памятью, емкость которой равна всему адресному пространству, определяемому размером адресных полей в форматах команд и базовых регистров.

Пользователь имеет в своем распоряжении все адресное пространство системы независимо от объема ее физической памяти и объемов памяти, необходимых для других программ, участвующих в мультипрограммной обработке. На всех этапах подготовки программ, включая загрузку в ОП, программа представляется в виртуальных адресах, и лишь при самом исполнении машинной команды производится преобразование виртуальных адресов в реальные адреса действующей памяти (в так называемые физические адреса.). Преобразование виртуальных адресов в физические упрощается и устраняется фрагментация памяти, если физическую и виртуальную память разбить на блоки двух типов. К первому

типу относятся блоки, образующие страницы, ко второму типу - сегменты.

При страничной организации память разбивается на страницы фиксированного размера, например, по 512 байт. В них осуществляется загрузка программ под управлением операционной системы. Страницам виртуальной и физической памяти присваивают номера, называемые номерами соответственно виртуальных и физических страниц. Каждая физическая страница способна хранить одну из виртуальных страниц. Порядок расположения (нумерация) байт в виртуальной и физической страницах сохраняется одним и тем же.

В мультипрограммной системе страничная организация памяти дает определенные преимущества. Когда новая программа загружается в ОП, она может быть направлена в любые свободные в данный момент физические страницы независимо от того, расположены они подряд или нет. Не требуется перемещения информации в остальной части памяти. Страничная организация позволяет сократить объем передачи информации между внешней памятью и ОП, так как страница программы не должна нагружаться до тех пор, пока она действительно не понадобится. Сначала в ОП загружается начальная страница программы и ей передается управление. Если по ходу работы делается попытка выборки слов из другой страницы, то производится автоматическое обращение к операционной системе, которая осуществляет загрузку из ВЗУ требуемой страницы.

На рис. 78 показано соответствие между виртуальной и физической памятью, устанавливаемое страничной таблицей. Очевидно, что физические страницы могут содержаться в текущий момент времени как в оперативной, так и во внешней памяти.

Страничная таблица для каждой программы формируется операционной системой в процессе распределения памяти и перерабатывается ею каждый раз, когда в распределении памяти производятся изменения. Процедура обращения к памяти состоит в том, что номер виртуальной страницы извлекается из адреса и используется для входа в страничную таблицу, которая указывает номер соответствующей физической страницы. Этот номер вместе с номером байта, взятым непосредственно из виртуального адреса,

представляет собой физический адрес, по которому происходит обращение к ОП.

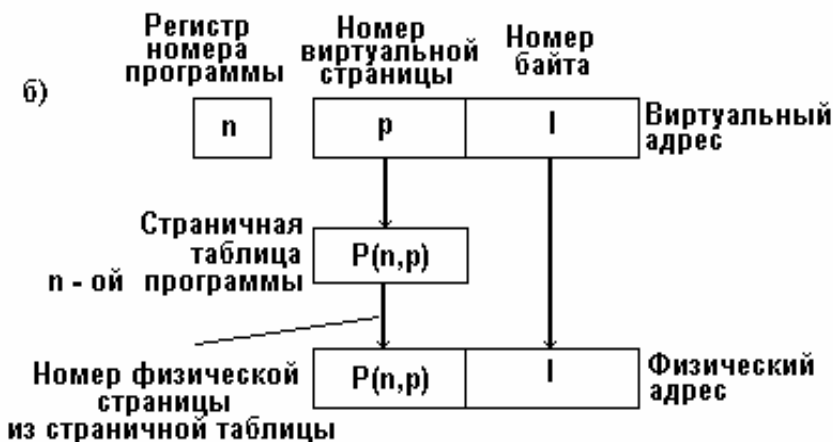


Рис.78

Если страничная таблица указывает на размещение требуемой информации во внешней памяти, то обращение к ОП не может состояться немедленно: операционная система должна организовать передачу из внешней памяти в ОП нужной страницы.

Для каждой из программ, обрабатываемых в мультипрограммном режиме, организуется своя область виртуальной памяти

и создается своя страничная таблица, при этом все программы делят между собой одну общую физическую память.

Страничные таблицы программ хранятся в ОП, и обращение к нужной строке активной страничной таблицы в ОП происходит по адресу, который определяется номером активной программы и номером виртуальной страницы.

Для ускорения преобразования адресов используется небольшая сверхоперативная память, куда передается из ОП страничная таблица активной программы. В другом варианте в сверхоперативной памяти могут находиться сведения о номерах виртуальных и соответствующих физических страниц для нескольких недавно использовавшихся страниц, в том числе принадлежащим разным программам. В этом варианте сверхоперативная память, используемая при преобразовании адресов, строится как ассоциативная с обращением не по адресу, а по содержанию хранимой в ячейке информации — в данном случае по хранимому в ячейке номеру программы и номеру виртуальной страницы.

Сегментная организация находит широкое применение в модульном программировании, при котором с целью упрощения понимания, написания и контроля программы и/или отдельных ее частей для реализации каждой функции используется отдельный программный модуль. Модуль определяется функцией, которую он выполняет, а не размерами, которые могут быть разными для различных модулей. Сегментная организация виртуальной памяти позволяет каждому модулю занимать свою собственную сплошную область памяти, тогда как при страничной организации модуль разбивается на страницы.

Логический адрес, вырабатываемый процессором для системы сегментной виртуальной памяти, состоит из номера сегмента и смещения. Таблица сегментов содержит значения базовых адресов (представляющих собой начальные адреса сегментов в физической памяти) и границ (определяющих объемы сегментов). Величина смещения не должна превышать значение границы. Защита сегмента, содержащего законченный программный модуль, обеспечивается путем однократного задания прав на использование и доступ для записи и чтения.

В процессе работы системы с сегментной виртуальной памятью в ней могут появляться так называемые «дырки», представляющие собой неиспользуемые области, которые требуют объединения. В системах реального времени это действие может служить причиной определенных неудобств. Сегментная виртуальная память должна быть снабжена алгоритмом размещения, с помощью которого осуществляется поиск неиспользуемых зон памяти для размещения каждого сегмента. В системах страничной виртуальной памяти необходимости в этих сложных алгоритмах нет; кроме того, при страничной организации виртуальной памяти отсутствует опасность появления «дырок» между программами (внешней фрагментации), а величина «дырок» внутри программы (внутренней фрагментации) не может превышать объема одного блока для каждой программы.

Сегментно - страничная организация памяти. До сих пор предполагалось, что виртуальная память, которой располагает программист, представляет собой непрерывный массив с единой нумерацией байт. Однако программа обычно состоит из нескольких массивов — подпрограмм, одной или нескольких секций данных. Так как заранее длины этих массивов неизвестны, то удобно, чтобы при программировании каждый массив имел свою собственную нумерацию байт, начинающуюся с нуля и продолжающуюся в возрастающем порядке. Желательно также, чтобы составленная таким образом программа могла работать при динамическом распределении памяти, не требуя от программиста усилий по объединению различных ее частей в единый массив. Эта задача решается в некоторых вычислительных системах путем использования особого метода преобразования виртуальных адресов в физические, называемого сегментно - страничной организацией памяти.

Виртуальная память каждой программы делится на части, именуемые сегментами, с независимой адресацией байт внутри каждой части. К виртуальному адресу следует добавить дополнительные разряды левее номере страницы; эти разряды определяют номер сегмента.

Возникает определенная иерархия в организации программ, состоящая из четырех ступеней: программа → сегмент → страница

→ байт. Этой иерархии программ соответствует иерархия таблиц, служащих для перевода виртуальных адресов в физические. Программная таблица для каждой программы, загруженной в систему, указывает начальный адрес соответствующей сегментной таблицы. Сегментная таблица перечисляет сегменты данной программы с указанием начального адреса страничной таблицы, относящейся к данному сегменту. Страничная таблица определяет расположение каждой из страниц сегмента в памяти. Страницы сегмента могут располагаться не подряд, часть страниц данного сегмента может находиться в оперативной памяти, остальные — во внешней.

Рассмотрим для примера организацию сегментно-страничной виртуальной памяти некоторой вычислительной системы. Пусть сегмент представляет собой блок последовательных адресов размером 64 Кбайт или 1 Мбайт, размер страницы - 2 или 4 Кбайт. При этом начальные адреса сегментов и страниц кратны их размерам. Размеры сегментов и страниц виртуальной памяти активной в данный момент программы задаются значениями «0» соответствующих разрядов управляющего регистра. Тогда виртуальный адрес (как и физический) будет иметь длину 24 разряда, причем поле номера сегмента будет занимать 8 или 4 старших разряда соответственно для сегментов размером в 64 Кбайт и 1 Мбайт, поле номера байта - 11 или 12 младших разрядов для страниц размером 2048 и 4046 байт. Промежуточные разряды адреса занимает поле номера страниц, которое может иметь 4, 5, 8 или 9 разрядов в зависимости от размеров сегмента и страницы.

Сегментные и страничные таблицы находятся в ОП, а в программной таблице нет необходимости, так как для каждой активной в данный момент программы управляющий регистр хранит начальный адрес и длину соответствующей сегментной таблицы. Хранит он также и номер программы.

Процесс преобразования адресов представлен на рис. 79. В общем случае преобразование адреса происходит в два этапа и требует двух дополнительных обращений к ОП (рис. 79, а). На первом этапе начальный адрес сегментной таблицы, установленный в управляющем регистре 1, суммируется с номером сегмента из виртуального адреса. В результате образуется адрес, по которо-

му из ОП считывается строка сегментной таблицы, содержащая адрес начала и длину страничной таблицы для данного сегмента. На втором этапе полученный адрес начала страничной таблицы суммируется с номером страницы из виртуального адреса, при этом образуется адрес, по которому из ОП считывается строка страничной таблицы. Если эта страница оказывается в ОП, то в старшие разряды регистра физического адреса передается ее номер, а в младшие заносится номер байта из регистра виртуального адреса. Формирование физического адреса на этом завершается.

Если нужная физическая страница оказывается во внешней памяти, то формируется сигнал прерывания, осуществляющий одноименную процедуру, называемую прерыванием по страничному сбою.

Операционная система инициирует передачу этой страницы из внешней памяти в ОП (при этом меняется номер физической страницы) и корректирует соответствующим образом страничную таблицу, находящуюся в оперативной памяти. В старшие разряды регистра физического адреса передается новый номер физической страницы, а в младшие — номер байта. Таким образом формируется физический адрес.

Далее выполняется запрошенное программой обращение к ОП. Одновременно информация о текущей странице (номерах программы, сегмента, виртуальной и соответствующей физической страницы) помещается в сверхоперативную ассоциативную память или в блок быстрой переадресации небольшой емкости.

Ассоциативная память. Ассоциативная память хранит указанные данные для небольшого числа недавно использовавшихся страниц. При наличии ассоциативной памяти значительно ускоряется процесс преобразования адресов, так как на каждом участке вычислительного процесса обращения к ОП сосредотачиваются на небольшом числе страниц, и поэтому всегда существует большая вероятность того, что текущее обращение произойдет к странице, информация о которой уже имеется в ассоциативной памяти, а следовательно, возможно быстрое преобразование адресов без дополнительных обращений к ОП.

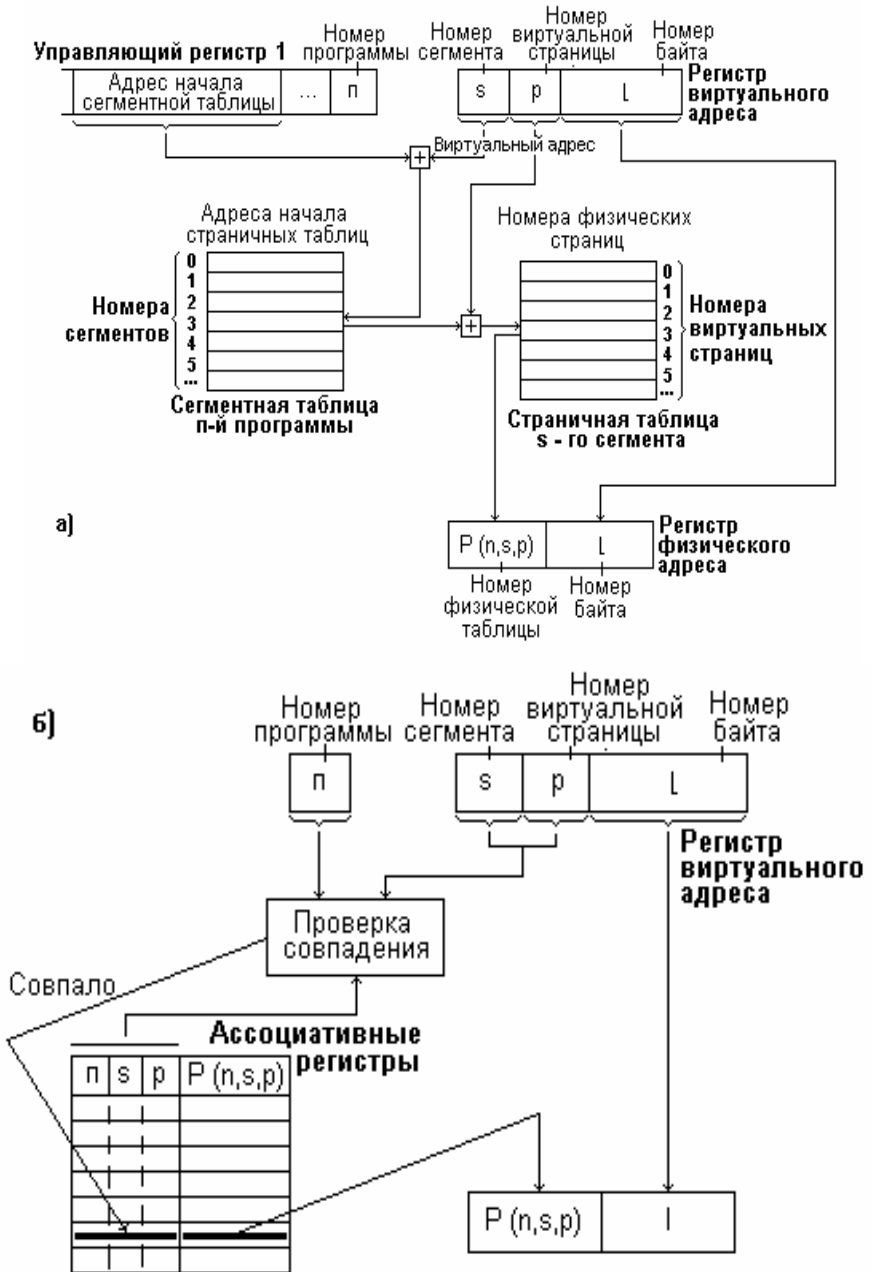


Рис.79.

Преобразование адресов всегда начинается с просмотра ассоциативной памяти. Если оказывается, что в одной из ее строк (ассоциативном регистре) хранится информация о странице, к которой должно произойти обращение, то из этой строки непосредственно выбирается номер физической страницы и дополнительные обращения к ОП (к сегментной и страничной таблицам) не производятся (рис.79,6). Если нужной информации нет в ассоциативной памяти, то делается попытка сократить время преобразования путем исключения одного дополнительного обращения к ОП (первый этап на рис. 79, а). Может оказаться так, что страница, к которой происходит обращение, принадлежит сегменту предыдущего обращения к ОП.

В аппаратуре преобразования адресов сохраняются номер сегмента и адрес начала его страничной таблицы для предыдущего обращения. Если совпадают номера сегментов текущего и предыдущего обращений, первый этап преобразования исключается, используется сохраненный адрес начала сегментной таблицы и выполняется только второй этап преобразования, т. е. производится только одно дополнительное обращение к ОП. Если номера сегментов не совпадут, то реализуется полная процедура преобразования адресов, показанная на рис. 79, а.

Дополнительные обращения к ОП сопровождаются занесением информации о текущей странице в ассоциативную память. Если в ассоциативной памяти не оказывается свободного регистра (строки), то данные о новой странице записываются на место данных, которые дольше других не использовались в процессе преобразования адресов.

13. Технологические аспекты полупроводниковой технологии

Совершенствование технологического процесса изготовления микропроцессоров – это главный атрибут повышения их быстродействия и надежности. Переход на новые техпроцессы является очевидным шагом, но технологам это дается каждый раз все с большим трудом. Современные процессоры выполняются по технологии 0,13 и 0,09 мкм, причем последняя была введена в 2004 году. Как видно, для этих техпроцессов соблюдается закон Мура,

который гласит, что каждые два года частота кристаллов удваивается при увеличении количества транзисторов с них. С такими же темпами сменяется и техпроцесс. Правда, в дальнейшем «гонка частот» опередит этот закон. К 2006 году компания Intel планирует освоение 65-нм техпроцесса, а 2009 – 32-нм

Здесь пора вспомнить структуру транзистора (рис.80), а именно - тонкий слой диоксида кремния SiO_2 , изолятора, находящегося между затвором и каналом, и выполняющего вполне понятную функцию - барьера для электронов, предотвращающего утечку тока затвора. Очевидно, что чем толще этот слой, тем лучше он выполняет свои изоляционные функции, но он является составной частью канала, и не менее очевидно, что если мы собираемся уменьшать длину канала (размер транзистора), то нам надо уменьшать его толщину, причем, весьма быстрыми темпами. За последние несколько десятилетий толщина этого слоя составляет в среднем порядка $1/45$ от всей длины канала. Но у этого процесса есть свой конец - как утверждал пять лет назад все тот же Intel, при продолжении использования SiO_2 , как это было на протяжении последних 30 лет, минимальная толщина слоя будет составлять 2.3 нм, иначе ток утечка тока затвора приобретет просто нереальные величины.

Для снижения подканальной утечки до последнего времени ничего не предпринималось. Сейчас ситуация начинает меняться, поскольку рабочий ток, наряду со временем срабатывания затвора, является одним из двух основных параметров, характеризующих скорость работы транзистора, а утечка в выключенном состоянии на нем непосредственно сказывается - для сохранения требуемой эффективности транзистора приходится, соответственно, поднимать рабочий ток, со всеми вытекающими условиями.

Изготовление микропроцессора - это самый сложный процесс, включающий более 300 этапов. Микропроцессоры формируются на поверхности тонких круговых пластин кремния - подложках, в результате определенной последовательности различных процессов обработки с использованием химических препаратов, газов и ультрафиолетового излучения.

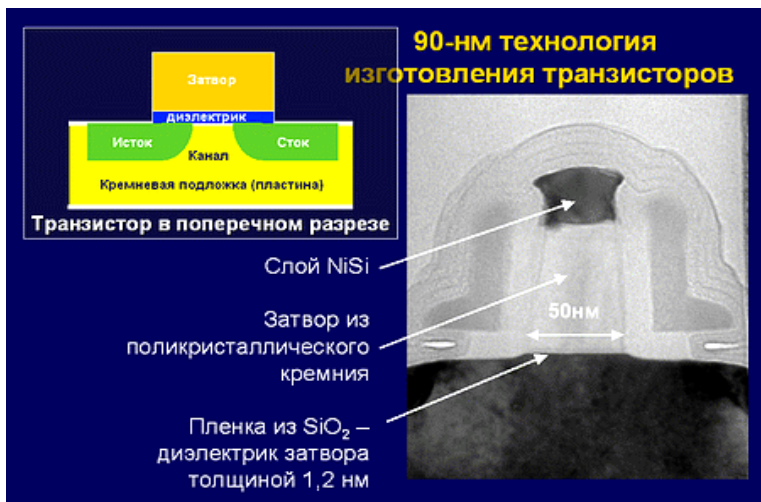


Рис. 80

Подложки обычно имеют диаметр 200 миллиметров, или 8 дюймов. Однако корпорация Intel уже перешла на пластины диаметром 300 мм, или 12 дюймов. Новые пластины позволяют получить почти в 4 раза больше кристаллов, и выход годных значительно выше. Пластины изготавливают из кремния, который очищают, плавят и выращивают из него длинные цилиндрические кристаллы. Затем кристаллы разрезают на тонкие пластины и полируют их до тех пор, пока их поверхности не станут зеркально гладкими и свободными от дефектов. Далее последовательно циклически повторяясь производят термическое оксидирование (формирование пленки SiO_2), фотолитографию, диффузию примеси (фосфор), эпитаксию (наращивание слоя).

В процессе изготовления микросхем на пластины-заготовки наносят в виде тщательно рассчитанных рисунков тончайшие слои материалов. На одной пластине помещается до нескольких сотен микропроцессоров, для изготовления которых требуется совершить более 300 операций. Весь процесс производства процессоров можно разделить на несколько этапов: выращивание диоксида кремния и создание проводящих областей, тестирование, изготовление корпуса и доставка.

Процесс производства микропроцессора начинается с "выращивания" на поверхности отполированной пластины изоляционного слоя диоксида кремния. Осуществляется этот этап в электрической печи при очень высокой температуре. Толщина оксидного слоя зависит от температуры и времени, которое пластина проводит в печи.

Затем следует фотолитография - процесс, в ходе которого на поверхности пластины формируется рисунок-схема. Сначала на пластину наносят временный слой светочувствительного материала – фоторезист, на который с помощью ультрафиолетового излучения проецируют изображение прозрачных участков шаблона, или фотомаски. Маски изготавливают при проектировании процессора и используют для формирования рисунков схем в каждом слое процессора. Под воздействием излучения засвеченные участки фотослоя становятся растворимыми, и их удаляют с помощью растворителя (плавиковая кислота), открывая находящийся под ними диоксид кремния.

Открытый диоксид кремния удаляют с помощью процесса, который называется "травлением". Затем убирают оставшийся фотослой, в результате чего на полупроводниковой пластине остается рисунок из диоксида кремния. В результате ряда дополнительных операций фотолитографии и травления на пластину наносят также поликристаллический кремний, обладающий свойствами проводника. В ходе следующей операции, называемой "легированием", открытые участки кремниевой пластины бомбардируют ионами различных химических элементов, которые формируют в кремнии отрицательные и положительные заряды, изменяющие электрическую проводимость этих участков.

Наложение новых слоев с последующим травлением схемы осуществляется несколько раз, при этом для межслойных соединений в слоях оставляются "окна", которые заполняют металлом, формируя электрические соединения между слоями. В своем 0.13-микронном технологическом процессе корпорация Intel применила медные проводники. В 0.18-микронном производственном процессе и процессах предыдущих поколений Intel применяла алюминий. И медь, и алюминий - отличные проводники электричества. При использовании 0,18-мкм техпроцесса использовалось 6 слоев, при

внедрении 90 нм техпроцесса в 2004 году применили 7 слоев кремния.

Каждый слой процессора имеет свой собственный рисунок, в совокупности все эти слои образуют трехмерную электронную схему. Нанесение слоев повторяют 20 - 25 раз в течение нескольких недель.

Чтобы выдержать воздействия, которым подвергаются подложки в процессе нанесения слоев, кремниевые пластины изначально должны быть достаточно толстыми. Поэтому прежде чем разрезать пластину на отдельные микропроцессоры, ее толщину с помощью специальных процессов уменьшают на 33% и удаляют загрязнения с обратной стороны. Затем на обратную сторону "похудевшей" пластины наносят слой специального материала, который улучшает последующее крепление кристалла к корпусу. Кроме того, этот слой обеспечивает электрический контакт между задней поверхностью интегральной схемы и корпусом после сборки.

После этого пластины тестируют, чтобы проверить качество выполнения всех операций обработки. Чтобы определить, правильно ли работают процессоры, проверяют их отдельные компоненты. Если обнаруживаются неисправности, данные о них анализируют, чтобы понять, на каком этапе обработки возник сбой.

Затем к каждому процессору подключают электрические зонды и подают питание. Процессоры тестируются компьютером, который определяет, удовлетворяют ли характеристики изготовленных процессоров заданным требованиям.

После тестирования пластины отправляются в сборочное производство, где их разрезают на маленькие прямоугольники, каждый из которых содержит интегральную схему. Для разделения пластины используют специальную прецизионную пилу. Не работающие кристаллы отбраковываются.

Затем каждый кристалл помещают в индивидуальный корпус. Корпус защищает кристалл от внешних воздействий и обеспечивает его электрическое соединение с платой, на которую он будет впоследствии установлен. Крошечные шарики припоя, расположенные в определенных точках кристалла, припаивают к

электрическим выводам корпуса. Теперь электрические сигналы могут поступать с платы на кристалл и обратно.

В будущих процессорах компания Intel применит технологию BVUL, которая позволит создавать принципиально новые корпуса с меньшим тепловыделением и емкостью между ножками CPU.

После установки кристалла в корпус процессор снова тестируют, чтобы определить, работоспособен ли он. Неисправные процессоры отбраковывают, а исправные подвергают нагрузочным испытаниям: воздействию различных температурных и влажностных режимов, а также электростатических разрядов. После каждого нагрузочного испытания процессор тестируют для определения его функционального состояния. Затем процессоры сортируют в зависимости от их поведения при различных тактовых частотах и напряжениях питания.

Известно, что существующие КМОП-транзисторы имеют много ограничений и не позволят в ближайшем будущем поднимать частоты процессоров также безболезненно. В конце 2003 года специалисты Intel сделали очень важное заявление о разработке новых материалов для полупроводниковых транзисторов будущего. Прежде всего, речь идет о новом диэлектрике затвора транзистора с высокой диэлектрической проницаемостью (так называемый «high-k»-материал), который будет применяться взамен используемого сегодня диоксида кремния SiO_2 (рис. 81), а также о новых металлических сплавах, совместимых с новым диэлектриком затвора. Решение, предложенное исследователями, снижает ток утечки в 100 раз, что позволяет вплотную подойти к внедрению производственного процесса с проектной нормой 45 нанометров. Оно рассматривается экспертами как маленькая революция в мире микроэлектронных технологий.

Чтобы понять, о чем идет речь, взглянем сначала на обычный МОП-транзистор, на базе которого делаются МП. В нем затвор из проводящего поликремния отделен от канала транзистора тончайшим (толщиной всего 1,2 нм или 5 атомов) слоем диоксида кремния (материала, десятилетиями используемого в качестве подзатворного диэлектрика).

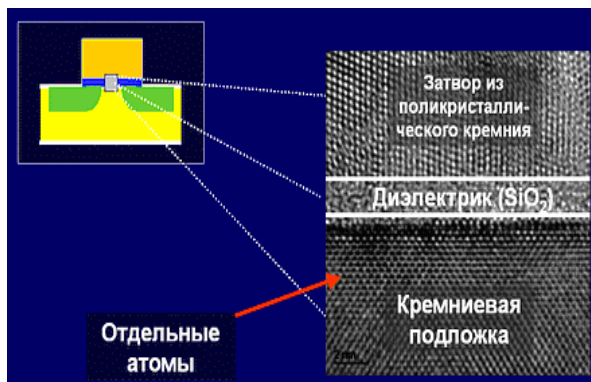


Рис. 81

Столь малая толщина диэлектрика необходима для получения не только малых габаритов транзистора в целом, но и для его высочайшего быстродействия (заряженные частицы передвигаются быстрее через затвор, в результате чего такой VT может переключаться до 10 миллиардов раз в секунду). Упрощенно - чем ближе затвор к каналу транзистора (то есть, чем тоньше диэлектрик), тем «большее влияние» в плане быстродействия он будет оказывать на электроны и дырки в канале транзистора.

Но с другой стороны, такой тонкий диэлектрик пропускает большие паразитные токи электронов утечки из затвора в канал (идеальный МОП-транзистор должен пропускать ток от истока к стоку и не пропускать - от затвора к истоку и стоку).

И в современных высокоинтегрированных микросхемах с сотнями миллионов транзисторов на одном кристалле токи утечки затворов становятся одной из фатальных проблем, препятствующих дальнейшему наращиванию количества транзисторов на кристалле. Более того, чем меньше по размерам мы делаем транзистор, тем тоньше нужно делать подзатворный диэлектрик. Но при его толщинах менее 1 нм резко (по экспоненте) возрастают туннельные токи утечки, что делает принципиально невозможным создание традиционных транзисторов менее определенных «горизонтальных» размеров (если при этом мы хотим получить от них хорошие скоростные характеристики). По оценкам экспертов, в

современных чипах почти 40% энергии может теряться из-за утечек.

Поэтому важность открытия ученых Intel нельзя недооценивать. После пяти лет исследований в лабораториях корпорации разработали специальный материал, позволяющий заменить традиционный диоксид кремния в обычном маршруте производства микросхем. Требования к такому материалу весьма серьезны: высокая химическая и механическая (на атомарном уровне) совместимость с кремнием, удобство производства в едином цикле традиционного кремниевого техпроцесса, но главное - низкие утечки и высокая диэлектрическая проницаемость.

Если мы боремся с утечками, то толщину диэлектрика нужно повысить хотя бы до 2-3 нм (рис. 82).

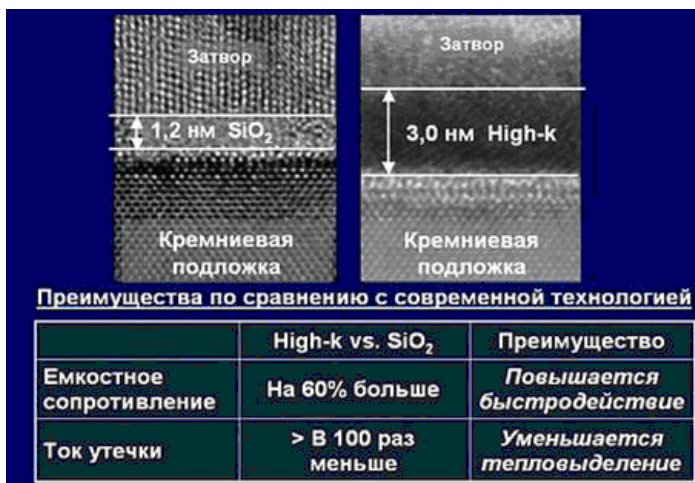


Рис. 82

Чтобы при этом сохранить прежнюю крутизну транзистора (зависимость тока от напряжения) необходимо пропорционально увеличить диэлектрическую проницаемость материала диэлектрика. Если проницаемость объемного диоксида кремния равна 4 (или чуть меньше в сверхтонких слоях), то разумной величиной диэлектрической проницаемости нового «интеловского» диэлектрика можно считать величину в районе 10-12. Несмотря на то, что материалов с такой диэлектрической проницаемостью немало (кон-

денсаторные керамики или монокристалл кремния), тут не менее важны факторы технологической совместимости материалов. Поэтому для нового high-k-материала был разработан свой высокоточный процесс нанесения, во время которого формируется один молекулярный слой этого материала за один цикл (рис. 83).

Исходя из этой картинки можно предположить, что новый материал - это тоже оксид. Причем монооксид, что означает применение материалов преимущественно второй группы, например, магния, цинка или даже меди.

Но диэлектриком дело не ограничилось. Потребовалось сменить и материал самого затвора - привычный поликристаллического кремния. Дело в том, что замена диоксида кремния на high-k-диэлектрик ведет к проблемам взаимодействия с поликристаллическим кремнием (ширина запрещенной зоны транзистора определяет минимально возможные для него напряжения).

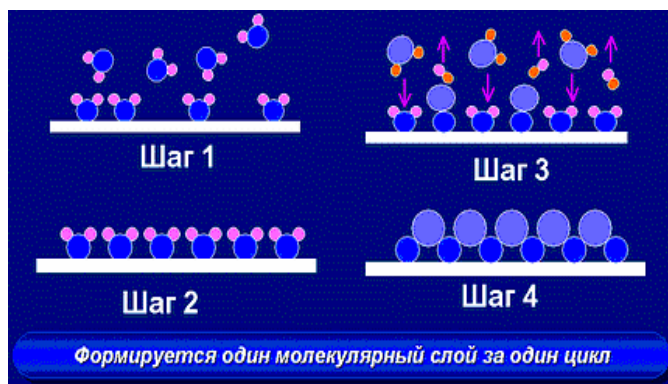


Рис. 83

Эти проблемы удастся устранить, если использовать специальные металлы для затворов транзисторов обоих типов (n-МОП и p-МОП) в сочетании с особым технологическим процессом. Благодаря этой комбинации материалов удастся достичь рекордной производительности транзисторов и уникально низких токов утечки, в 100 раз меньших, чем при использовании нынешних материалов (рис. 84).



Рис. 84

В этом случае уже не возникает искушения использовать для борьбы с утечками значительно более дорогую технологию SOI (кремний на изоляторе), как это делают некоторые крупные производители микропроцессоров.

Отметим также еще одно технологическое новшество Intel - технологию напряженного (strained) кремния, которая впервые используется в 90-нанометровых процессорах Prescott и Dothan. Наконец-то, компания Intel в подробностях рассказала, каким именно образом происходит формирование слоев напряженного кремния в ее КМОП-структурах. КМОП-ячейка состоит из двух транзисторов - n-МОП и p-МОП (рис. 85).

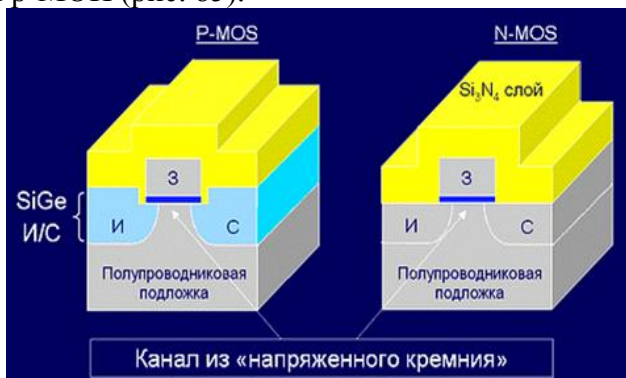


Рис. 85

В первом (n-MOS) канал транзистора (n-канал) проводит ток при помощи электронов (отрицательно заряженных частиц), а во втором (p-MOS) - при помощи дырок (условно положительно заряженных частиц). Соответственно, и механизмы формирования напряженного кремния у этих двух случаев различны. Для n-MOS-транзистора используется внешнее покрытие слоем нитрида кремния (Si_3N_4), который за счет механических напряжений немного (на доли процента) растягивает (в направлении протекания тока) кристаллическую решетку кремния под затвором, в результате чего рабочий ток канала возрастает на 10% (условно говоря, электронам становится более просторно двигаться в направлении канала). В p-MOS-транзисторах все наоборот: в качестве материала подложки (точнее - только областей стока и истока) используется соединение кремния с германием (SiGe), что немного сжимает кристаллическую решетку кремния под затвором в направлении канала. Поэтому дыркам становится «легче» «передвигаться» сквозь акцепторные атомы примеси, и рабочий ток канала возрастает на 25%. Сочетание же обеих технологий дает 20-30-процентное усиление тока.

Таким образом, применение технологии «напряженного кремния» в обоих типах устройств (n-MOS и p-MOS) приводит к значительному повышению производительности транзисторов при повышении себестоимости их производства всего лишь на ~2% и позволяет создавать более миниатюрные транзисторы следующих поколений. В планах Intel - использовать напряженный кремний для всех будущих техпроцессов вплоть до 22-нанометрового.

Материал с низкой диэлектрической проницаемостью используется в качестве диэлектрика медных соединений (рис. 86) во всех техпроцессах Intel, начиная с 0,13-микронного. Он уменьшает величину паразитной емкости, которая возникает между медными соединениями на кристалле, что повышает скорость передачи внутренних сигналов и уменьшает энергопотребление. Intel - первая и пока единственная компания, которая использует этот low-k-материал для изоляции межсоединений.

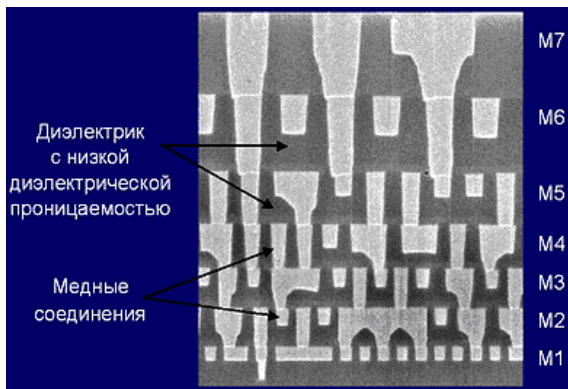


Рис. 86

Приложение 1. История развития компьютеров – период до появления первого ПК

- 1617 год. Джон Непер (John Napier) создал деревянную машину для выполнения простейших вычислений.
- 1642 год. Блез Паскаль (Blaise Pascal) описал машину для суммирования чисел.
- 1822 год. Чарльз Баббадж (Charles Babbage) представил механическое устройство, названное позднее аналитической машиной, которую можно считать первой настоящей вычислительной машиной.
- 1906 год. Ли Ди Форест (Lee DeForest) запатентовал вакуумный триод, использовавшийся в качестве переключателя в первых электронных компьютерах.
- 1945 год. Джон фон Нейман (John Von Neumann) написал статью First Draft of a Report on the EDVAC, в которой была описана архитектура современных программируемых компьютеров.
- 1946 год. Джоном Мошли (John Mauchly) и Дж. Преспером Экертом (J. Presper Eckert) создана электронно-вычислительная машина ENIAC.
- 1947 год. 23 декабря Джоном Бардином (John Bardeen), Уолтером Браттейном (Walter Brattian) и Уильямом Шокли (William Shockley) был успешно протестирован первый транзистор, совершивший переворот в полупроводниковой технике.

- 1949 год. В Кембриджском университете Морис Вилке (Maurice Wilkes) создал первый практический программируемый компьютер EDSAC.

- 1950 год. Исследовательская организация в Миннеаполисе представила первый коммерческий компьютер ERA 1101.

- 1952 год. В U.S. Census Bureau был установлен компьютер UNIVAC I.

- 1953 год. Фирма IBM создала первый электронный компьютер 701.

- 1954 год. Впервые появился в продаже полупроводниковый транзистор стоимостью 2.5 доллара, созданный Гордоном Тилом (Gordon Teal) в фирме Texas Instruments, Inc.

- 1954 год. Фирма IBM выпустила первый массовый калькулятор 650; в течение этого же года было продано 450 экземпляров данной модели.

- 1955 год. Фирма Bell Laboratories анонсировала первый транзисторный компьютер TRADIC

- 1956 год. В Массачусетском технологическом институте создан первый многоцелевой транзисторный программируемый компьютер TX-0.

- 1956 год. С появлением модели IBM 305 RAMAC начинается эра устройств магнитного хранения данных.

- 1958 год. Джек Килби (Jack Kilby), сотрудник фирмы Texas Instruments, создает первую интегральную схему, состоящую из транзисторов и конденсаторов на одной полупроводниковой пластине.

- 1959 год. Фирма IBM создает серию мэйнфреймов 7000 - первых транзисторных компьютеров для крупных компаний.

- 1959 год. Роберт Нойс (Robert Noyce) - компании Fairchild Camera и Instruments Corp. - создает интегральную схему с помощью расположения соединительных каналов непосредственно на кремниевой пластине.

- 1960 год. В фирме DEC создан первый миникомпьютер PDP-1, стоимостью 120 тыс. долларов.

- 1961 год. По данным журнала Datamation, продукция фирмы IBM занимала 81,2% компьютерного рынка; в этом году IBM анонсирована серию систем 1400.

- 1964 год. Суперкомпьютер CDC 6600, созданный Сеймуром Креем (Seymour Cray) выполнял около 3 млн инструкций в секунду, что в три раза больше, чем у его ближайшего конкурента IBM Stretch.

- 1964 год. Фирма IBM анонсировала семейство компьютеров System/360 (шесть совместимых модификаций и 40 периферийных устройств).

- 1964 год. Впервые в мире была проведена транзакция в реальном времени на системе IBM SABRE.

- 1965 год. Фирма Digital Equipment Corporation анонсировала первый успешный коммерческий проект мини-компьютера PDP-8.

- 1966 год. Фирма Hewlett Packard представила компьютер для бизнеса HP-2115, который по производительности не уступал большим корпоративным системам.

- 1970 год. Впервые в мире осуществлена связь между двумя компьютерами; первые четыре узла сети ARPAnet - университет Калифорнии, UCLA, SRI International и университет штата Юта.

- 1971 год. В лаборатории фирмы IBM в Сан-Хосе создана 8-дюймовая дискета.

- 1971 год. В журнале Electronic News впервые появилась реклама микропроцессоров Intel 4004

- 1971 год. В журнале Scientific American впервые появилась реклама одного из первых персональных компьютеров Kenback-1 стоимостью 750 долларов.

- 1972 год. Фирма Hewlett Packard представила систему HP-35 с постоянной памятью.

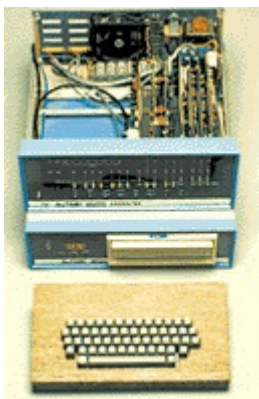
- 1972 год. Дебют микропроцессора Intel 8008.

- 1973 год. Роберт Меткалф (Robert Metcalfe) описал метод сетевого соединения Ethernet в исследовательском центре Пало Альто фирмы Хегох.

- 1973 год. Фирма Micral выпустила первый коммерческий персональный компьютер на основе микропроцессора Intel 8008.

- 1973 год. Дон Ланкастер (Don Lancaster) создал на основе телевизионного приемника первый буквенно-цифровой монитор TV Typewriter.

- 1974 год. Компания Micro Instrumentation Telemetry Systems (MITS), занимающаяся электроникой в городе Альбукерке (шт. Нью-Мексико) объявила о разработке небольшого компьютера для индивидуального пользования. Эд Робертс и двое его партнеров создали небольшой сборный компьютер. Он получил название Altair.



- 1974 год. Scelbi (SCientific ELectronic and Biological) Computer Consulting представила машину на базе процессора Intel — 8008. Она имела 1 кбайт программируемой памяти и была предназначена в основном для научного применения.



- 1974 год. В исследовательском центре Пало Альто фирмы Хегох создана рабочая станция, в качестве устройства ввода которой использовалась мышь.

- 1975 год. Появилась первая коммерческая сеть с пакетной коммутацией Telnet - гражданский аналог сети ARPAnet.

- 1975 год. В январском выпуске журнала Popular Electronics описан компьютер Altair - 8800, созданный на базе процессора Intel 8080.



- 1976 год. Стив Возняк (Steve Wozniak) создал одноплатный компьютер Apple I



- 1976 год. Фирмой Shugart Associates анонсирован первый 5,25-дюймовый гибкий диск и дисковод.

- 1976 год. Создан первый коммерческий векторный процессор Cray I

- 1977 год. Фирма Tandy Radio Shack выпустила компьютер TSR-80.

- 1977 год. Создан компьютер Apple II.



- 1977 год. Фирма Commodore выпустила компьютер PET (Personal Electronic Transactor).

- 1978 год. Фирмой Digital Equipment Corporation создан компьютер VAX 11/780, способный адресовать 4,3 Гбайт виртуальной памяти.

- 1979 год. Фирма Motorola выпустила микропроцессор 68000.

- 1980 год. Джон Шох (John Shoch) из исследовательского центра Пало Альто фирмы Хегох обнаружил первого компьютерного "червя" - небольшую программу, которая распространялась в сети в поиске свободных процессоров.

- 1980 год. Seagate Technologies выпустила первый жесткий диск для микрокомпьютеров.

- 1980 год. Разработан первый оптический диск, емкость которого в 60 раз превышала емкость 5,25-дюймового гибкого диска.

- 1981 год. Адам Осборн (Adam Osborne) выпустил первый портативный компьютер Osborne I стоимостью 1 795 долларов.



- 1981 год. Фирма IBM выпустила свой первый персональный компьютер PC.

- 1981 год. Фирма Sony анонсировала первую 3,5-дюймовую дискету и дисковод.
- 1983 год. Фирма Apple выпустила компьютер Lisa с первым графическим интерфейсом пользователя.



- 1983 год. Фирма Compaq Computer Corp. выпустила первый клон компьютера IBM PC.



- 1984 год. Фирма Apple стала выпускать первый самый успешный компьютер с графическим интерфейсом пользователя, который принес 1,5 млн долларов только за этот год.



- 1984 год. Фирма IBM создала компьютер PC-AT на базе процессора Intel 286.



- 1985 год. Выпущен первый музыкальный компакт-диск и накопитель CD-ROM.

- 1986 год. Фирма Compaq выпустила компьютер Deskpro 386, в котором впервые был установлен процессор Intel 386.

- 1987 год. Фирма IBM приступила к производству компьютеров семейства PS/2, в которых был установлен 3,5-дюймовый дисковод и VGA-видеоадаптер.

- 1988 год. Один из основателей Apple Стив Джобс (Steve Jobs) покидает эту фирму и создает собственную компанию NeXT.

- 1988 год. Фирма Compaq и другие производители PC - совместимых систем разработали новую, улучшенную архитектуру компьютера.

- 1988 год. Роберт Моррис (Robert Morris) создает и запускает своего "червя" в сеть ARPAnet; заражено по различным оценкам от 6 000 до 60 000 узлов.
- 1989 год. Фирма Intel выпускает процессор 486, который содержит 1 млн транзисторов.
- 1990 год. В Женеве в исследовательском центре CERN разработан язык разметки гипертекста (Hypertext Markup Language - HTML) и на свет появилась World Wide Web (WWW).
- 1993 год. Фирма Intel выпустила первый процессор Pentium из семейства P5. Кроме выпуска процессора, Intel разработала для него набор микросхем системной логики.
- 1995 год. Фирма Intel начала продавать процессор Pentium Pro - первого представителя семейства P6.
- 1995 год. Компания Microsoft представила первую 32-разрядную операционную систему Windows 95.
- 1997 год. Фирма Intel выпустила процессор Pentium II, построенный на базе Pentium Pro с поддержкой инструкции MMX.
- 1998 год. Компания Microsoft анонсировала новую версию своей операционной системы Windows 98.
- 1998 год. Фирма Intel представляет процессор Celeron - более дешевую версию Pentium II.
- 1999 год. Фирма Intel выпустила процессор Pentium III, построенный на базе Pentium II с поддержкой инструкции SSE (Streaming SIMD Extensions).
- 2000 год. Компания Microsoft выпустила операционную систему Windows 2000.
- 2000 год. Фирмы Intel и AMD объявили о выпуске процессоров с тактовой частотой 1 ГГц.
- 2000 год. Фирма Intel анонсирует процессор Itanium - первый процессор семейства P7.



Приложение 2. История развития МП Intel

"Intel - это процессоры"- Энди Гроув (Andy Grove). До начала 70-х годов вычислительные машины были доступны весьма ограниченному кругу специалистов, а их применение, как правило, оставалось окутанным завесой секретности и мало известным широкой публике. Однако в 1971 г. произошло событие, которое в корне изменило ситуацию и с фантастической скоростью превратило компьютер в повседневный рабочий инструмент десятков миллионов людей. В том вне всякого сомнения знаменательном году еще почти никому не известная фирма Intel из небольшого американского городка с красивым названием Санта-Клара (шт. Калифорния) создала новый полупроводниковый прибор, получивший название "МИКРОПРОЦЕССОР".

Сегодня, оглядываясь на 25 лет назад, мы не очень погрешили бы против истины, добавив к этому слову титул "Его Величество", - столь велико оказалось влияние крошечного полупроводникового кристалла практически на все сферы деятельности. Именно ему мы обязаны появлением нового класса вычислительных систем - персональных компьютеров, которыми теперь пользуются, по существу, все: от учащихся начальных классов и бухгалтеров до маститых ученых и инженеров. Этим машинам, не занимающим и половины поверхности обычного письменного стола, покоряются все новые и новые классы задач, которые ранее были доступны (а по экономическим соображениям часто и недос-

тупны - слишком дорого тогда стоило машинное время мэйн-фреймов и мини-ЭВМ) лишь системам, занимавшим не одну сотню квадратных метров. Наверное, никогда прежде человек не имел в своих руках инструмента, обладающего столь колоссальной мощностью при столь микроскопических размерах.

Начало В 1968 г. Гордон Мур (Gordon Moore) и Боб Нойс (Bob Noyce), одни из тех, кто закладывал фундамент известной полупроводниковой компании Fairchild Semiconductor, основали фирму Intel Corporation. Первой идеей нового предприятия было создание полупроводниковых запоминающих устройств, призванных заменить ЗУ на магнитных сердечниках. Поскольку к концу 60-х годов память этого типа практически исчерпала весь свой потенциал развития, проблема была весьма актуальной, а ее разработка сулила немалые прибыли. И хотя в данной области Intel добилась заметных успехов, тем не менее мировую славу ей принесли совсем другие изделия. Поворотным моментом в истории компании стал 1969 г., когда был получен заказ на создание ряда специализированных микросхем для калькуляторов от ныне уже несуществующей японской фирмы Busicom. В апреле того же года в штаб-квартиру Intel прибыли три инженера из Busicom, среди которых был идеолог нового проекта Масатоси Шима (Masatoshi Shima).

Через несколько лет этому человеку суждено будет сыграть одну из главных ролей в создании кристалла i8080, во многом предопределившего дальнейший путь развития корпорации Intel. Выполнение заказа японской компании поручили Марциану Хоффу (Marcian Hoff), в то время ведущему специалисту Intel. Проанализировав техническое задание (ТЗ), а также предложенный Шима вариант архитектуры и системы команд, Хофф пришел к выводу, что поставленную задачу можно решить и более простыми средствами, оптимизировав систему команд и дополнив схему устройства модулем памяти. Окончательная версия ТЗ была им разработана совместно с другим инженером компании, Стэном Мэзором (Stan Mazor), и предусматривала функционирование процессора на тактовой частоте 1 МГц. Любопытно, что на такой же частоте работал и компьютер модели 1620 фирмы IBM, но при этом ориентировочная стоимость изделия Intel составляла 30 - 40 дол., а одна

только арендная плата за пользование IBM 1620 доходила до 2000 дол. в месяц. Темп работы над прибором был настолько высок, что уже осенью 1969 г. заказчику представили проект кристалла. Поскольку вариант, предложенный Intel, оказался более универсальным и имел потенциал для применения не только в калькуляторах, Busicom отдала предпочтение американской разработке. Для японского менеджмента того времени подобное решение было беспрецедентным, особенно если принять во внимание молодость компании из Соединенных Штатов - в 1969 г. фирме Intel исполнился всего лишь год.

Воплотить идеи Хоффа в кремнии выпало Федерико Фэггину (Federico Faggin), который и осуществил это менее чем за год. Кстати, спустя всего лишь несколько лет он станет одним из основателей и президентом фирмы Zilog, которая получит широкую известность благодаря микропроцессору Z80, и поныне выпускаемому рядом изготовителей. Вместе с тем, несмотря на успешный ход работ, одно обстоятельство беспокоило многих проектировщиков прибора. Согласно контракту, Busicom имела на кристалл исключительные права, но специалисты, его создавшие, прекрасно сознавали, что возможностей чипа вполне достаточно для гораздо более широкого круга применений, нежели только калькуляторы. Поэтому, когда в начале 1971 г. Busicom была вынуждена вступить с Intel в переговоры о снижении цен на поставляемые микросхемы (в связи с обострением конкуренции на рынке калькуляторов), Хофф рекомендовал сотрудникам отдела маркетинга согласиться на это в обмен на право продажи кристалла на открытом рынке. Японская сторона приняла предложение Intel.

Сразу после этого Эд Гелбах (Ed Gelbach), специалист по маркетингу компании, и его помощник Хэнк Смит (Hank Smith) приступили к исследованию потенциального рынка сбыта новых микросхем. Первые полученные результаты вполне обнадеживали. В ходе опроса потенциальных потребителей удалось установить, что стоимость - единственный фактор, удерживающий проектировщиков от программирования логических функций в их оборудовании. А на вопрос: "Будете ли вы делать это, если цена составит 5 дол.?" - был получен столь же краткий, сколь и категоричный ответ: "Безусловно". Резюме маркетинговому исследованию

подвел Боб Нойс: "Все, что нам надо - это довести стоимость комплекта интегральных схем до 30 или 40 дол., и тогда покупатели начнут писать собственные программы. Следует создать спрос и обеспечить приемлемую цену". В процессе исследования рынка стал очевиден еще один немаловажный фактор - потенциальным потребителям кристаллов необходимо оказать помощь в их применении.

Это натолкнуло команду Гелбаха на идею создания систем проектирования микроЭВМ, содержащих по крайней мере простейшие инструментальные средства разработки и отладки программного обеспечения. Идея оказалась настолько плодотворной, что уже через несколько лет поставки подобных систем стали приносить доход, соизмеримый с доходом от продажи самих микропроцессоров. В результате кристалл вышел на рынок в сопровождении соответствующих средств поддержки и началось триумфальное шествие микропроцессоров по всему миру. 15 ноября 1971 г. можно считать началом новой эры в электронике. В этот день компания приступила к поставкам первого в мире микропроцессора Intel 4004 - именно такое обозначение получил первый прибор, послуживший отправной точкой абсолютно новому классу полупроводниковых устройств.

Кристалл представлял собой 4-разрядный процессор с классической архитектурой ЭВМ гарвардского типа и изготавливался по передовой в те годы r-канальной МОП-технологии с проектными нормами 10 мкм. Электрическая схема прибора насчитывала 2300 транзисторов. Микропроцессор работал на тактовой частоте 750 кГц при длительности цикла команды 10,8 мкс. Чип i4004 имел адресный стек (счетчик команд и три регистра стека типа LIFO - Last In First Out), блок регистров общего назначения - РОН (регистры сверхоперативной памяти, или регистровый файл), 4-разрядное параллельное АЛУ, аккумулятор, регистр команд с дешифратором команд и схемой управления, а также схему связи с периферийными устройствами. Все эти функциональные узлы объединялись между собой 4-разрядной шиной данных (рис. 1). Для однокристалльного процессора i4004 имел весьма впечатляющие характеристики. Память команд достигала 4Кбайт (для сравнения: объем ЗУ мини-ЭВМ в начале 70-х годов редко превышал

16 Кбайт), а регистровый файл ЦП насчитывал шестнадцать 4-разрядных регистров, которые можно было использовать и как восемь 8-разрядных (восемь 4-разрядных пар). Такая организация РОНов сохранена и в последующих микропроцессорах фирмы Intel. Три регистра стека обеспечивали три уровня вложения подпрограмм. Конечно, эта цифра не вызвала особых восторгов у программистов, тем не менее они получили возможность создавать полноценные программы. Процессор i4004 монтировался в пластмассовый или металлокерамический корпус типа DIP (Dual In-line Package) всего с 16 выводами.

В систему его команд входило 46 инструкций. По своему функциональному составу она была универсальной, т. е. рассчитана на широкий круг решаемых задач и разрабатываемых приложений. Первоначальное назначение кристалла наложило определенный отпечаток на состав системы команд, поэтому присутствие в ней ряда инструкций, в частности десятичной коррекции, а также наличие соответствующих аппаратных средств не вызывает особого удивления.

Вместе с тем кристалл располагал весьма ограниченными средствами ввода/вывода, а в системе команд отсутствовали операции логической обработки данных (И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ), в связи с чем их приходилось реализовывать с помощью специальных подпрограмм, что в некоторых случаях чрезмерно усложняло создаваемое ПО. Модуль i4004 не имел возможности останова (команды HALT) и обработки прерываний. Впрочем, это объясняется не упущением создателей устройства, а тем, что в калькуляторах, где поначалу и планировалось использовать прибор, особой необходимости в этих средствах нет. Цикл команды процессора состоял из восьми тактов задающего генератора. Такое их количество вызывает удивление, но объясняется очень просто, хотя с позиций сегодняшнего дня и несколько неожиданно. Как уже отмечалось, чип i4004 монтировался в корпус всего с 16 выводами - самый распространенный (а значит, и самый дешевый) тип корпуса в начале 70-х годов. А поскольку в распоряжении инженеров оказался узкий интерфейс с "внешним миром", то пришлось пойти на применение мультиплексированной шины адреса и данных, причем 12-разрядный адрес выдавать порциями по четыре

разряда, что, конечно, не могло не сказаться на длительности машинного цикла. Прием команды по такому интерфейсу требовал еще двух тактов. На исполнение же самой инструкции из восьми тактов процессор затрачивал лишь три.

Таким образом, соотношение "накладные расходы/полезная работа" составило 5:3 в пользу накладных расходов. Узкое "окно" во внешний мир долгое время было бичом всех микропроцессоров без исключения. Забегая несколько вперед, можно сказать, что 40-выводной корпус во многом решил проблемы 8-разрядных систем, но уже первые 16-разрядные приборы опять поставили на повестку дня этот больной вопрос. Задача создания многовыводных корпусов оказалась крепким орешком и попортила немало крови конструкторам и технологам, не говоря уж о самих разработчиках процессоров, которых к тому времени число "40", наверное, приводило просто в бешенство.

Примерно к середине 70-х годов появились корпуса типов DIP и QIP (QUad In-line Package), имевшие до 64 выводов, но и им по ряду причин не суждено было стать панацеей от всех бед. Во-первых, рост степени интеграции и тактовых частот БИС не могли не привести к увеличению потребляемой мощности. Пластмассовые корпуса позволяли рассеивать мощность не более 1,5 - 2 Вт. Металлокерамика увеличивала этот показатель до 3 - 4 Вт, но одновременно поднимала стоимость микросхемы на такую высоту, что ее массовый выпуск сразу оказывался под вопросом. Во-вторых, 64 вывода "спасали" разработчиков процессоров на крайне ограниченном временном отрезке. Как бы там ни было, но спрос рождает предложение и проблема корпусов с большим числом выводов и приемлемой ценой была решена в начале 80-х годов. Однако все это было впереди, а в самом начале 70-х пионерам пришлось исходить из существовавших реалий и принимать соответствующие технические решения. Говоря о первом в мире микропроцессоре, нельзя не вспомнить и о том, что с самого начала специалисты позаботились о простоте и удобстве построения систем на базе i4004. Компанией был разработан и выпущен не один кристалл центрального процессора, а целое семейство БИС, в которое вошли ПЗУ 4001, ОЗУ 4002, регистр сдвига 4003 и ряд других вспомогательных микросхем. Поскольку все они были рассчитаны

на совместное использование, разработка аппаратных средств системы заметно упрощалась, и это стало не последней причиной популярности i4004. $i4004 + \text{обработка прерываний} + \dots = i4040$.

Опыт использования первого МП показал, что такие факторы, как отсутствие средств обработки прерываний, наличие трех уровней вложения подпрограмм и необходимость реализации логических операций И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ с помощью специальных подпрограмм, далеко не всегда удовлетворяют разработчиков. Создав новый рынок и захватив на нем господствующие высоты, Intel тем не менее стремилась расширить его границы, и решение о выпуске усовершенствованного варианта i4004, свободного от указанных недостатков, было вполне логичным ходом компании. Использование же в новом кристалле всех наработок, приобретенных при создании предыдущего процессора, дало существенную экономию времени, и неудивительно, что i4040 вышел в свет вскоре после появления i4004.

В новом приборе сохранены все функциональные возможности предшествующей модели и существенно улучшены как технические, так и программные средства. Система команд пополнилась 14 инструкциями, включая выполнение логических операций И и ИЛИ; кроме того, в процессор были введены средства останова и обработки прерываний. Претерпела некоторые изменения и архитектура устройства. Адресный стек процессора увеличен с трех до семи регистров, а количество РОНов возросло с 16 до 24, причем их разбили на две области, выбираемые при помощи специальных команд. Отчасти такая организация обусловлена тем, что процессор теперь мог обращаться к двум блокам памяти команд объемом 4 Кбайт и за каждым из них программист мог закрепить свою область регистров. Наряду с этим восемь РОНов были всегда доступны для использования. В итоге получилась достаточно гибкая и удобная структура, позволявшая разрабатывать самостоятельные программные модули, способные взаимодействовать через общую часть регистрового файла. Обработка одноуровневых прерываний - одно из наиболее существенных новшеств чипа i4040 - превратила его в полноценный процессор и сделала возможным использование в системах реального масштаба времени. Благодаря применению сигнала "останов" стала реальностью

синхронизация работы процессора с некоторыми внешними событиями.

Вместе с тем специальных операций со стеком (запись в стек содержимого РОН и извлечение его из стека) разработчики ПО так и не дождалось. Несмотря на то что тактовая частота и машинный цикл i4040 не претерпели изменений, производительность процессора возросла за счет использования более совершенной архитектуры и эффективной системы команд. 60 инструкций, ориентированных на широкий спектр решаемых задач, обработка прерываний, до 8 Кбайт памяти команд, а также возможность быстрого перевода систем на базе i4004 на новый процессор вывели i4040 в безусловные лидеры рынка 4-разрядных устройств. Правда, полной совместимости с программным обеспечением, разработанным под i4004, получить не удалось, и некоторые программы пришлось переписывать.

Безусловно, необходимость доработки ПО при переводе систем с ЦП i4004 на новый процессор не вызвала особой радости у специалистов, но и не разочаровала их настолько, чтобы они отвергли новое детище Intel. Преимущества, которые i4040 имел перед своим предшественником, с лихвой компенсировали отмеченный недостаток. Кроме того, в то время проблема переносимости ПО еще не успела стать актуальной: создатели процессора i4040 не ощущали того давления со стороны потребителей своей продукции, которое они начнут испытывать спустя всего 5 - 7 лет, а потому могли себе позволить такую роскошь, как частичная несовместимость моделей процессоров в рамках одного семейства. Принимая во внимание особенности архитектуры и технические характеристики кристалла, i4040 можно считать первым прибором второго поколения 4-разрядных систем. Вот только век этого поколения оказался чрезвычайно коротким - новым процессором фирма Intel закрыла линию 4-разрядных устройств.

Следующая высота - 8 разрядов. Выпустив на рынок первый микропроцессор, Intel не стала изменять привычкам лидера и 1 апреля 1972 г. начала поставки первого в отрасли 8-разрядного прибора Intel 8008. Создание этого чипа весьма напоминает историю с i4004. Как и 4-разрядный микропроцессор, он был разработан для нужд другой компании, только в этом случае заказчиком выступа-

ла американская фирма Computer Terminals Corporation of Texas, позднее известная как Datapoint. Кристалл проектировали четыре человека, трое из которых (Хофф, Фэггин и Мэзор) уже имели опыт создания МП i4004. Четвертым членом команды стал новый сотрудник компании Хэл Фини (Hal Feeney). Проектирование i8008 шло практически параллельно с работами над i4004. Когда же процессор был готов, заказчик от него отказался, мотивируя это тем, что прибор получился слишком медленным для решения поставленных перед ним задач, а также требовал для своей работы большого количества вспомогательных микросхем. Тогда Intel решила пустить i8008 в свободную продажу и в очередной раз не ошиблась. Кристалл изготавливался по r-канальной МОП-технологии с проектными нормами 10 мкм и содержал 3500 транзисторов. Процессор работал на частоте 500кГц при длительности машинного цикла 20 мкс (10 периодов задающего генератора).

В отличие от своих предшественников новый МП имел архитектуру ЭВМ принстонского типа, а в качестве памяти допускал применение комбинации ПЗУ и ОЗУ. Помимо увеличения разрядности и перехода на использование общего поля памяти для команд и данных, структура процессора претерпела еще ряд существенных изменений. Прежде всего это коснулось регистрового файла и устройства управления. По сравнению с i4004 число РОНов уменьшилось вдвое (с 16 до 8), причем два регистра в основном использовались для хранения адреса при косвенной адресации памяти. В связи с этим следовало бы ожидать снижения производительности, которого на самом деле не произошло, поскольку операции с памятью i8008 выполнял быстрее предыдущих моделей благодаря меньшему количеству состояний в машинном цикле и отсутствию необходимости исполнения минимум трех подготовительных команд (как в i4004 и i4040) при обращении к ОЗУ или ПЗУ.

Вместе с тем, объем блока регистров был ограничен и возможностями технологии, которая в то время еще не позволяла размещать на кристалле большие регистровые структуры (аналогично кристаллам i4004 и i4040 в МП i8008 блок РОНов был реализован в виде динамической памяти, необходимость регенерации

которой влечет за собой применение ряда дополнительных аппаратурных средств).

Почти вдвое (с восьми до пяти состояний) сократилась длительность машинного цикла. Теперь процессор выполнял команды за один - три машинных цикла, а некоторые инструкции - за один цикл из трех состояний. Для синхронизации работы ЦП с медленными устройствами был введен сигнал готовности (READY). Разработчики технических средств на базе i8008 не были ограничены жесткими требованиями в отношении быстродействия микросхем памяти и периферийных устройств и могли использовать те ИС, которые наиболее полно соответствовали конкретной системе. В ряде случаев это приводило к ощутимому сокращению стоимости оборудования. Такой гибкостью 4-разрядные кристаллы похвастаться не могли.

Система команд первого 8-разрядного ЦП насчитывала 65 инструкций, причем значительно увеличилось число команд условных переходов, а также логических инструкций и команд сдвига. Новый кристалл мог адресовать память объемом до 16 Кбайт (объем ЗУ для ЦП типа i4040 не превышал 8 Кбайт). Его производительность по сравнению с 4-разрядными системами возросла в 2,3 раза. Процессор с такими параметрами уже можно было рассматривать как серьезную заявку на многие очень перспективные секторы рынка, включая контрольно-испытательное оборудование, прецизионную измерительную технику и сложные промышленные контроллеры систем управления технологическими процессами.

Однако i8008 получил от своих предшественников и "тяжелое наследство". Объем и организация стека остались такими же, как и у чипа i4040, и реализация операций с ним по-прежнему возлагалась на программиста. Узкий интерфейс с "внешним миром" ограничил количество управляющих сигналов процессора: в результате специалистам Intel пришлось использовать их шифрацию, что повлекло за собой необходимость установки дополнительного внешнего оборудования для формирования сигналов управления. В среднем для сопряжения процессора с памятью и устройствами ввода/вывода требовалось около 20 схем средней степени интеграции.

Если спортсмены исповедуют древний олимпийский принцип "Citius, altius, fortius", то разработчикам он ближе в несколько измененном виде: "Citius, citius, citius". Вскоре после выхода чипа Intel 8008 появилась его усовершенствованная версия i8008-1. Модернизированный вариант работал уже на частоте 800 кГц при длительности машинного цикла 12,5 мкс.

Увеличение в 1,5 раза производительности центрального процессора, наряду с большим (по тому времени) объемом оперативной памяти, послужило лучшей рекомендацией для активного использования кристалла в различных областях, начиная от промышленности и медицины и кончая военной электроникой и торговлей. По мере расширения сферы влияния микропроцессора и усложнения систем на его базе возросли и требования к нему со стороны проектировщиков оборудования. Программное обеспечение уже с трудом вписывалось в 16 Кбайт, да и производительность прибора начинала "поджимать" многих разработчиков. Кроме того, некоторые области применения настойчиво требовали расширения не только количества, но и номенклатуры периферийных устройств. Системщики уже с трудом могли обходиться без такой традиционной для мэйнфреймов и мини-ЭВМ периферии, как дисплеи, принтеры, накопители на магнитной ленте и дисках и т. п. Стало очевидно, что технические характеристики изделия превратились в фактор, сдерживающий его дальнейшее распространение. Гордон Мур, один из основателей и первый президент Intel, в те годы писал, что микропроцессоры найдут еще более широкое применение, если улучшить их рабочие характеристики, ибо уже выявились области, где требуются предельно высокие параметры.

Возможности р-канальной МОП-технологии для создания сложных высокопроизводительных МП были уже практически исчерпаны, поэтому направление главного удара перенесли на технологию n-МОП. Перед проектировщиками стояли не менее сложные проблемы - разработка эффективной системы команд, рассчитанной на широкий круг решаемых задач, при сохранении программной совместимости с предыдущей моделью, расширение объема адресуемой памяти, поддержка интенсивного ввода/вывода без существенной потери производительности процессора, совер-

шенствование подсистемы обработки прерываний. i8080 - триумф 8-разрядных систем. Работа над новым 8-разрядным процессором началась практически сразу после завершения опытно-конструкторского цикла, связанного с выпуском кристалла i8008, и некоторое время оба проекта шли практически параллельно. Костяк команды проектировщиков составили ставшие уже ветеранами "микропроцессорного фронта" Мэзор и Фэггин, а также бывший сотрудник японской фирмы Busicom Масатоси Шима, знакомый нам по кристаллу i4004.

Первоначально идея создания i8080 сводилась к повышению производительности чипа-предшественника только за счет перехода на новый технологический процесс. К этому времени Intel располагала технологией n-МОП, которая прошла обкатку на кристаллах памяти 2102. Стремясь сократить сроки проектирования, разработчики попытались использовать маски i8008 в производстве чипов ОЗУ 2102. Однако, как вспоминает Стэн Мэзор, после предварительного изучения вопроса стало ясно, что применить старые маски к новому технологическому процессу не удастся и их придется разрабатывать заново. Такой поворот событий привел к идее коренной модернизации схемы процессора и повышению его производительности примерно на порядок за счет сочетания преимуществ новой технологии и усовершенствованной архитектуры, включая расширенную систему команд. На проведение всего комплекса работ потребовалось более года, и 1 апреля 1974 г. (ровно через два года после выпуска i8008) микропроцессор Intel 8080 был представлен вниманию всех заинтересованных лиц.

Поставки начались по цене 360 дол. за кристалл. Дэйв Хаус (Dave House), один из ветеранов компании, так комментирует эту цифру: "По сути дела, прибор представлял собой настоящий компьютер, который стоил в то время тысячи долларов. Поэтому мы чувствовали, что предложили вполне разумную цену". По свидетельству Эда Гелбаха, расходы на исследования и разработки по программе i8080 окупились в течение первых пяти месяцев продаж. Новый ЦП практически по всем статьям разительно отличался от своих предшественников. Благодаря использованию технологии n-МОП с проектными нормами 6 мкм, на кристалле удалось разместить 6 тыс. транзисторов. При этом геометрические размеры

самого кристалла по сравнению с i8008 увеличились незначительно. Следовательно, процент выхода годных изделий и ряд экономических показателей производства, включая себестоимость, удалось сохранить на достаточно высоком уровне. Тактовая частота процессора была доведена до 2 МГц, что в 2,5 раза превышало аналогичный параметр для i8008, а длительность цикла команды составила уже 2 мкс.

Несмотря на чисто внешнее сходство структур i8080 и i8008, схема нового процессора существенно отличалась от предшествующей модели. К великой радости системщиков и программистов объем памяти, адресуемой процессором, был увеличен в четыре раза и достиг 64 Кбайт (кстати, в то время ОЗУ такой емкости предлагали потребителям минимальные конфигурации многих мини-ЭВМ). В сочетании с эффективным механизмом обработки прерываний это давало им карт-бланш для широкого применения нового МП в сложных системах сбора и обработки информации различного назначения, особенно функционирующих в реальном масштабе времени. За счет использования корпуса с 40 выводами удалось разделить адресную и информационную шины процессора, в результате отпала необходимость применения дополнительных внешних схем для разделения потоков адресов и данных. Общее же количество микросхем, требовавшихся для построения системы в минимальной конфигурации, сократилось с 20 до 6, т. е. более чем в три раза.

В регистровый файл были введены указатель стека, активно используемый при обработке прерываний, а также два программно-недоступных регистра для внутренних пересылок. Поскольку микропроцессор i8008 успешно продавался уже в течение двух лет и за этот период для него был наработан достаточно большой объем ПО, сохранение программной совместимости i8080 и i8008 было вполне естественным и разумным шагом компании (таким образом, Intel встала на тот же путь, что и корпорация ИВМ с компьютерами знаменитой серии System 360, оказавшись на долгие годы своего рода заложницей собственного творения). Именно поэтому в состав РОНов нового процессора были включены основные рабочие регистры предыдущей модели. Правда, полной совместимости с i8008 достичь опять не удалось, так как процедуры обраще-

ния к подпрограммам и инструкции ввода/вывода МП i8080 в значительной степени отличались от соответствующих процедур и операций кристалла i8008, и при переводе систем со старого процессора на новый в некоторых случаях программы приходилось полностью перерабатывать. Включение в систему команд ряда инструкций, адресующих память с использованием трех пар регистров (в i8008 для этого выделялась одна пара), придало дополнительную гибкость системе и существенно упростило жизнь программистам, реализация же блока РОНов на основе статической, а не динамической памяти дала дополнительную экономию площади кристалла для размещения других схем процессора. Исключение аккумулятора из регистрового файла и введение его в состав арифметико-логического устройства упростило схему управления внутренней шиной, поскольку при этом отпала необходимость в ее использовании для передачи данных между сверхоперативной памятью и АЛУ во время выполнения арифметических и логических операций.

Новым веянием в архитектуре микропроцессоров стало использование многоуровневой системы прерываний по вектору. Такое техническое решение позволило довести общее число источников прерываний в системе до 256. Правда, до появления специализированных БИС контроллеров прерываний схема формирования векторов прерываний требовала применения до десяти дополнительных чипов средней степени интеграции. В отличие от прерываний по вектору, размещение стека в оперативной памяти не было последним словом в архитектуре МП, но и здесь Intel не обошлась без "изюминки", добавив в схему микроЭВМ всего один триггер, в качестве стека можно было использовать отдельную память емкостью до 64 Кбайт, сэкономив тем самым ОЗУ для размещения программ и данных. Тот факт, что разработчики микропроцессоров воспользуются техническими решениями, которые уже нашли применение в мэйнфреймах и мини-ЭВМ, ни у кого не вызывал сомнений. Вопрос был лишь в том, что именно будет использовано и кто станет первым. Пионером в этом опять оказалась Intel.

Освобождение центрального процессора от управления внешними устройствами и обмен данными между памятью систе-

мы и периферией, минуя ЦП, были уже достаточно давно и успешно реализованы в универсальных ЭВМ (IBM System 360 и др.). Таким образом, появление в кристалле i8080 механизма прямого доступа к памяти при работе с внешними устройствами можно смело считать первым (но далеко не последним) ударом микропроцессоров по большим системам. ПДП открыл зеленую улицу для применения в микроЭВМ таких сложных устройств, как накопители на магнитных дисках и лентах, а также дисплеи на ЭЛТ, которые и превратили микроЭВМ в полноценную вычислительную систему.

Традицией компании, начиная с первого кристалла, стал выпуск не отдельного чипа ЦП, а семейства БИС, рассчитанного на совместное использование. Помимо микропроцессора, в новый набор микросхем вошли ИС системных генератора и контроллера. Вскоре их дополнили БИС контроллера ПДП и контроллера прерываний. Благодаря хорошо продуманному составу комплекта, проектирование микро-ЭВМ на его базе в ряде случаев упростилось настолько, что было подобно сборке домика из детских кубиков. Мимо такой техники пройти было трудно! Мощная система команд, высокое быстродействие, простота проектирования, прототипные комплекты и системы разработки вскоре превратили микропроцессор в стандарт де-факто. Более того, конкуренты отстали настолько, что около года Intel практически безраздельно господствовала на рынке. Однако спрос на ее кристаллы оказался чрезвычайно высок и не всегда удовлетворялся своевременно, так что для конкурентов появилась определенная ниша. Ряд полупроводниковых компаний, включая гигантов вроде Texas Instruments, купили у Intel лицензии на выпуск микропроцессоров, в результате возник институт так называемых вторых поставщиков. На первый взгляд могло показаться, что этот шаг отбирает прибыли у самой компании, на самом деле он только способствовал распространению ее продукции и укреплению позиций на рынке.

И все-таки рано или поздно любой монополии приходит конец. Высокие темпы роста рынка 8-разрядных систем не могли не привести к появлению на нем новых действующих лиц. Полупроводниковые компании, выпускавшие микропроцессоры собственной разработки, росли как грибы после дождя, и не было прак-

тически ни одной мало-мальски уважающей себя фирмы, которая не попробовала бы свои силы на этой стезе. Безусловно, все они отбирали у Intel определенную долю рынка, но самыми серьезными конкурентами стали Motorola и Zilog с кристаллами M6800 и Z80 соответственно (см. врезку "Motorola и Zilog против..."). Эти компании имели весьма серьезные намерения, и через некоторое время корпорации Intel пришлось потесниться.

Кристаллы от Motorola и Zilog обладали определенными преимуществами. Архитектура M6800 была более прозрачна для программистов, чем продукция Intel. Разработчики Z80 учли все недостатки микросхемы i8080, так как знали их лучше всех (и президент компании Федерико Фэггин, и ряд ведущих специалистов ранее работали в Intel и были не последними людьми в команде, создавшей МП i8080, что даже послужило основанием для обращения Intel в суд). Кроме того, для работы как M6800, так и Z80 требовался всего один номинал питания, а для конкурирующего изделия - три.

На вызов, брошенный фирмами Motorola и Zilog, компания Intel ответила разработкой серии периферийных контроллеров, которые существенно упростили построение сложных систем, а также ЦП i8085, уже свободного от указанных недостатков, имевшего ряд преимуществ перед конкурентами и обладавшего программной совместимостью со своим предшественником. Но, пожалуй, самым сильным ответным ходом компании стало создание системного ПО - однопользовательской ОС ISIS II и ОС реального времени iRMX-80. Мощь этого удара оказалась столь велика, что многие компании так и не смогли от него оправиться, а некоторым для восстановления былых рыночных позиций потребовался не один год и не один миллиард долларов. Intel обеспечила свои ЦП (а значит и потребителей) мощнейшей программной поддержкой, которую в то время больше никто не смог предложить. Это предредило исход борьбы, и последствия той победы компания, по видимому, ощущает до сих пор. Вторая половина 70-х годов была отмечена массивной атакой японских поставщиков полупроводниковых устройств на рынок США. Объемы производства американских компаний росли, а прибыли падали. Пожалуй, наибольшему давлению подвергся сектор ЗУ, которые занимали не

последнюю строку в списке приоритетов Intel. Но и к такому повороту событий фирма оказалась готовой, приступив в 1976 г. (к немалому удовольствию изготовителей законченных систем) к выпуску одноплатных микроЭВМ серии iSBC на базе своих микропроцессорных комплектов.

Операционная система iRMX-80 стала для этих машин базовым системным ПО. Таким образом, из изготовителя микросхем корпорация превратилась в производителя подсистем, обеспечив себе тем самым дополнительный стабильный рынок сбыта полупроводниковых изделий. Именно с тех пор в лексикон "кремниевых" компаний прочно вошел термин OEM (Original Equipment Manufacturer - изготовитель комплексного оборудования).

Ориентацию на перспективу можно смело отнести к характерной особенности стиля Intel Corporation. Вероятно, нельзя назвать ни одного прибора, работа над которым не проводилась бы одновременно с проектированием изделия следующего поколения (достаточно вспомнить историю создания кристалла i8008). Следуя установившейся традиции, в конце 1974 г. фирма приступила к созданию прототипа 32-разрядной системы iAPX-432, на которую возлагались большие надежды. Рост производительности планировалось получить не только благодаря увеличению разрядности, но и за счет использования сложной архитектуры с возможностью организации мультипроцессорной системы.

Следует отметить, что мультипроцессорные универсальные ЭВМ в ту пору еще не стали привычным атрибутом вычислительных центров и многие нюансы построения таких машин были известны весьма ограниченному кругу специалистов. Сыграло ли это свою роль или имелись другие обстоятельства - сегодня судить сложно, но работа над новым ЦП шла не так быстро, как хотелось. В то же время, несмотря на ряд удачных действий, давление на рынке 8-разрядных приборов со стороны главных конкурентов Motorola и Zilog хотя и ослабло, но продолжало оставаться объективной реальностью. Динамика выполнения проекта, связанного с iAPX-432, давала все основания полагать, что выпуск на рынок 32-разрядных процессоров будет задержан, а это означало потерю темпа. Вакуум, который мог образоваться в связи с проблемами 432-го кристалла, требовалось срочно заполнить. Поэтому руково-

дство компании сочло целесообразным попытаться развить успех i8080, и в самом начале 1976 г. стартовали работы по созданию 16-разрядного прибора, который впоследствии получил обозначение i8086.

Конечной целью нового проекта было получение 16-разрядного микропроцессора с производительностью, на порядок превышающей аналогичный параметр кристалла i8080. Поставленная задача решалась за счет дальнейшего совершенствования архитектурных концепций, положенных в основу его предшественника. Избранная стратегия эволюционного, а не революционного развития (как в случае iAPX-432) оказалась верной и скоро дала свои плоды. Менеджером работ был назначен Жан -Клод Корне (Jean Claude Cornet), занимавший в то время пост технического директора компании по микропроцессорам. Коллектив, ядро которого составили Боб Кохлер (Bob Koehler), Джон Бэйлис (John Bayliss), Джим Маккевит (Jim McKeivitt), Чак Уайлдмэн (Chuck Wildman) и Стив Морз (Steve Morse), возглавил Билл Полмэн (Bill Pohlman).

Время и давление конкурентов были критическими факторами, и очень скоро численность команды увеличилась почти до 20 человек. По тем временам это выглядело весьма необычно. Но, пожалуй, еще любопытнее было то, что многие из них имели опыт работы не более года. Тем не менее они сумели создать сложный и исключительно удачный процессор в очень сжатые сроки. Новый кристалл был анонсирован 8 июня 1978 г. (через 2,5 года после начала выполнения проекта!). Прибор изготавливался по высококачественной трехмикронной МОП-технологии с кремниевыми затворами (H-MOS), позволившей разместить на чипе 29 тыс. транзисторов. Высокое быстродействие элементов (задержка 2 нс/вентиль) обеспечило тактовую частоту процессора 5 МГц, а 16-разрядная архитектура и 200-нс машинный цикл - производительность процессора, превышающую аналогичный параметр i8080 на порядок величины.

Программная совместимость с i8080/i8085 была, пожалуй, единственной, но вместе с тем и исключительно важной характеристикой, которая объединяла 86-й кристалл с его предшественниками. Структура процессора оказалась полностью пересмотрен-

ной. Прежде всего, прибор был разбит на два функциональных блока (рис. 3) - операционный (Execution Unit, EU) и интерфейсный (Bus Interface Unit, BIU), которые могли работать одновременно. В результате исполнение одной команды совмещалось во времени с выборкой следующей инструкции или данных из памяти. Более того, в ЦП появился небольшой буфер команд, что давало дополнительную экономию времени при обращениях к памяти. Так, микропроцессоры позаимствовали у универсальных ЭВМ еще одно техническое решение, и именно с этого момента в них началась реализация принципов параллелизма.

Возможность адресации 1-Мбайт ОЗУ и сегментация памяти могут быть отнесены к одним из наиболее существенных новшеств, предложенных инженерами Intel. В частности, сегментация памяти и большое число уровней прерываний были ориентированы на работу систем в многозадачном режиме, весьма актуальном для приложений управления. (Правда, следует отметить, что механизм защиты памяти реализован не был, и в ряде случаев это существенно усложняло разработку ПО). Большая емкость ОЗУ позволяла перевести проекты построения сложных операционных и прикладных систем из области теории в сферу практической реализации.

Наряду с поддержкой ввода/вывода по каналу прямого доступа к памяти чип i8086 обеспечивал адресацию до 64К портов программно-управляемого ввода/вывода. Это снимало практически любые ограничения при формировании крупных систем сбора и обработки информации. Исключительно удачной находкой можно считать два режима работы процессора - минимальный и максимальный. Первый рассчитан на использование ЦП в небольших системах и предполагал работу кристалла без БИС контроллера шины. Максимальный режим был ориентирован на применение чипа в сложных крупномасштабных системах и требовал наличия указанного контроллера. Таким образом, один и тот же процессор с одинаковым успехом мог применяться в системах различного класса. Конкуренты же для этих целей выпускали разные модификации своих продуктов.

Однако, пожалуй, наиболее интересна система команд процессора. 147 инструкций позволяли решать задачи управления практически

любой сложности. Появление среди них таких операций, как умножение и деление 16-разрядных чисел со знаком и без знака, команд обработки массивов данных, а также программно-управляемых прерываний дает все основания назвать этот кристалл универсальным, рассчитанным на использование не только в сложных контроллерах, но и в качестве центрального процессора ЭВМ общего назначения.

Как и всегда, новый прибор вышел в мощном сопровождении средств поддержки: вспомогательных БИС, средств разработки и отладки аппаратуры и системного ПО, а также прототипных комплектов и одноплатных микроЭВМ серии iSBC86.

Несмотря на блестящие характеристики продукта и мощную поддержку со стороны изготовителя, потребовалось почти два года, чтобы кристалл завоевал признание разработчиков. Первое время даже его создатели стали опасаться - не слишком ли сложное детище они сотворили. Однако феномену медленного роста объема продаж нашлось разумное толкование: начальные темпы и не могли быть высокими, поскольку создаваемой системе требуется новое системное и прикладное ПО. Справедливость такого объяснения подтвердилась в начале 1980 г., когда закончился лабораторный период отработки компьютеров на базе ЦП i8086 и они вышли на стадию промышленного производства - кривая сбыта стала резко подниматься. Intel начала занимать на рынке 16-разрядных систем господствующие высоты.

Вместе с тем конкуренты тоже не сидели сложа руки. Компания Motorola извлекла уроки из битвы за сектор 8-разрядных систем и спустя примерно год после появления чипа i8086 предложила очень удачный микропроцессор M68000 (см. врезку "На рынке 16-разрядных систем..."). К концу 1979 г. его присутствие на рынке ощущалось весьма явственно. Некоторые сотрудники верхнего эшелона управления корпорации Intel пришли к выводу, что, если не сделать собственный кристалл стандартным прибором для отрасли, это может привести к потере рынка следующего поколения МП. Поэтому усилия были сосредоточены на ускорении выпуска процессора с 8-разрядной внешней шиной данных (i8088) и реализации плана агрессивного маркетинга приборов, получившего название Operation Crush. В основу кампании, которую воз-

главил Билл Дэвидову (Bill Davidow), были положены два момента: во-первых, поставка пользователю готовых решений, включая системное ПО (ОС iRMX-86), широкий набор периферийных БИС и оперативную техническую поддержку; во-вторых, подготовка потребителей к будущим техническим решениям на базе платформы Intel. В соответствии с новой стратегией претерпел изменения и традиционный рекламный девиз компании, вполне закономерно превратившись из Intel delivers ("Intel поставляет") в Intel delivers solutions ("Intel поставляет решения").

В рамках Operation Crush от Стокгольма до Сиэтла фирма провела множество семинаров по различным аспектам практического использования микропроцессорного комплекта i8086. Просветительская деятельность сопровождалась массивной рекламной кампанией. Акция имела грандиозный успех - в течение года процессор i8086 стал базовым элементом более 2 тыс. проектов. IBM PC - новая эпоха в истории развития Intel В начале 1980 г. IBM приступила к реализации проекта создания персонального компьютера. Новая машина должна была представлять собой открытую систему и базироваться на стандартном для отрасли микропроцессоре.

Работы шли в г. Бока-Ратон (шт. Флорида). К этому моменту на рынке присутствовали кристаллы i8086, i8088, Z8000 и M68000. Сотрудники Intel с замиранием сердца ждали решения "Голубого гиганта". Дэйв Хаус, тогда генеральный менеджер компании по микропроцессорам и периферийным кристаллам, вспоминает, что выбор IBM в пользу кристалла i8088 как центрального процессора для ПК стал очевиден после поступившего от дилера Intel из Бока-Ратона сообщения о начале выполнения заказов на системы разработки и внутрисхемные эмуляторы ICE-88. Но тем не менее в течение еще пяти месяцев Intel не могла сделать никаких официальных заявлений на этот счет. Поскольку Apple выбрала для своих машин кристалл M68000, успокаиваться было рано и кампания Operation Crush была продолжена. И только после получения от IBM первых крупных заказов на процессоры стало окончательно ясно - это самая крупная победа Intel. Теперь уже вне всякого сомнения именно ее приборы стали истинным стандартом в электронной промышленности. В 1984 г. объемы продаж МП

i8086 в девять раз превысили соответствующий показатель для M68000.

Использование микросхем i8086 в IBM PC предопределило дальнейшее развитие корпорации Intel как разработчика и изготовителя универсальных процессоров общего назначения. Вычислительная мощность 16-разрядных приборов была поддержана арифметическим сопроцессором i8087, который позволил превратить ПК в достаточно мощный инструмент и для решения задач вычислительного характера. Более того, теперь и разработчики систем управления на базе 86-го ЦП получили возможность использовать интенсивную арифметическую обработку информации, для которой ранее служили мини-ЭВМ. Бурный рост объемов продаж персональных компьютеров явно свидетельствовал о том, что рождается новый рынок огромной емкости.

В конце 70 - начале 80-х годов основными сферами применения микропроцессоров продолжали оставаться различные системы управления (ПК еще не появились). Сложность решаемых задач требовала высокой производительности ЦП, большого объема ОЗУ и поддержки многозадачных сред, характерных для подобного рода комплексов. Для разработчиков программного обеспечения 1 Мбайт оперативной памяти достаточно быстро превратился в своеобразное прокрустово ложе, а отсутствие в i8086 аппаратных средств поддержки защиты памяти существенно усложняло создание ПО. Решение этих задач и было основной целью разработки следующего кристалла.

Процессор i80286 был анонсирован 1 февраля 1982 г. и сразу приковал к себе пристальное внимание специалистов. Архитектура и характеристики чипа оказались весьма впечатляющими. Оставшись 16-разрядным прибором, по производительности новый ЦП в 3 - 6 раз превзошел своего предшественника при тактовой частоте первой модификации 8 МГц. Благодаря использованию многовыводного корпуса (к тому времени проблема их массового производства была успешно решена) разработчики смогли применить схему с отдельными шинами адресов и данных. 24 разряда адреса позволили обращаться к физической памяти объемом до 16 Мбайт - такую же емкость имели и старшие модели большинства мэйнфреймов. Встроенная система управления памя-

тью и средства ее защиты открывали широкие возможности использования МП в многозадачных средах. Кроме того, аппаратура i80286 обеспечивала работу с виртуальной памятью объемом до 1 Гбайт. Для поддержки устройства управления памятью система команд пополнилась еще 16 инструкциями.

Новый ЦП имел два режима работы - реальный и защищенный. В первом случае он воспринимался как быстрый ЦП i8086 с несколько расширенной системой команд и прекрасно подходил тем потребителям, для которых, помимо скоростных характеристик, жизненно важным было сохранение существующего задела ПО. Работа в защищенном режиме позволяла использовать преимущества прибора в полном объеме, и прежде всего - большой объем основной памяти. Этими качествами решила воспользоваться IBM, применив процессор в новой модели ПК типа AT, что с невероятной скоростью превратило i80286 в бестселлер рынка 16-разрядных систем. Как видим, очередной передел этого сектора, начавшийся в 1982 г., компания встретила во всеоружии. Помимо 286-го, в арсенале Intel были еще две модели процессоров i80186 и i80188 - модифицированные варианты микросхем i8086 и i8088 соответственно.

Достигнутые в технологии успехи позволили разработчикам разместить на кристалле контроллеры прерываний и ПДП, а также таймер и системный генератор. Таким образом, покупателю были предложены мощные однокристалльные ЭВМ, с помощью которых можно было реализовать законченную систему с числом ИС не более десяти. Уже в первый год после начала поставок количество проданных чипов i80186 превысило в 30 раз аналогичный показатель для i8086.

Спрос на i80286 также требовал постоянного увеличения объемов его производства. Длина слова - 32 разряда Использование корпорацией IBM микропроцессоров i80286 в новой модели персонального компьютера IBM PC/AT и стремительный рост объемов ее продаж стимулировали усилия специалистов Intel по разработке прибора следующего поколения. Увеличение тактовой частоты 286-го процессора сверх достигнутого предела в 16 МГц давалось уже слишком дорого, а кроме того, никак не устраняло узкого места системы, которым оставалась оперативная память.

Помимо прочего, 286-й решил далеко не все проблемы, характерные для многозадачных сред. У инженеров Intel было два пути кардинального повышения производительности процессора - 32-разрядная обработка данных и совершенствование тракта процессор - память. Эффективное же функционирование МП под управлением многозадачных ОС требовало усовершенствования устройства управления памятью. Таким образом, специалисты имели перед собой четкие задачи, которые необходимо было решить. (Любопытно, что примерно в это же время в американской прессе промелькнули сообщения о намерениях "Голубого гиганта" приобрести крупный пакет акций Intel и даже приводились оценки его размера, доходившие до 40%. Судя по всему, сделка не состоялась, но сам факт появления подобного рода информации свидетельствует о достаточно тесных контактах обеих компаний. Не исключено, что корпорация IBM выступала в качестве заказчика кристалла следующего поколения, а ее инженеры принимали непосредственное участие в формировании технических требований на процессор.)

Первенец 32-разрядных систем i80386 был представлен 17 октября 1985 г. и имел все права на звание процессора для ЭВМ общего назначения. Использование КМОП-технологии с проектными нормами 1 мкм и двумя уровнями металлизации позволило разместить на кристалле 275 тыс. транзисторов и реализовать полностью 32-разрядную архитектуру ЦП (рис.4). 32 разряда адреса обеспечили адресацию физической памяти объемом до 4 Гбайт и виртуальной памяти емкостью до 64 Тбайт. Встроенная в МП система управления памятью и защиты включала регистры преобразования адреса, механизмы защиты оперативной памяти и улучшенные аппаратные средства поддержки многозадачных ОС. Помимо работы с виртуальной памятью допускались операции с памятью, имевшей страничную организацию. Предварительная выборка команд, буфер на 16 инструкций, конвейер команд и аппаратная реализация функций преобразования адреса значительно уменьшили среднее время выполнения команды. Благодаря этим архитектурным особенностям, процессор мог выполнять 3 - 4 млн команд в секунду, что примерно в 6 - 8 раз превышало аналогичный показатель для МП i8086. Безусловно, новый прибор остался

совместимым со своими предшественниками на уровне объектных кодов. Одной из наиболее любопытных особенностей рассматриваемой разработки компании было использование кэш-памяти, позволившей существенно повысить производительность систем на базе 386-го процессора (еще один атрибут универсальных машин, который стал применяться в микропроцессорных системах).

Для управления работой буфера Intel создала БИС высокопроизводительного контроллера кэш-памяти типа i82385, с помощью которой формировался двухходовой множественный ассоциативный кэш. Указанная БИС обеспечивала управление буфером емкостью до 32 Кбайт и высокий коэффициент удачных обращений. Поскольку степень интеграции была еще недостаточна для реализации на том же кристалле и математического сопроцессора, он по-прежнему выпускался в виде отдельного кристалла i80387, дополняя вычислительную мощь ЦП.

Особый интерес представляли три режима работы кристалла - реальный, защищенный и режим виртуального МП i8086. В первом обеспечивалась совместимость на уровне объектных кодов с устройствами i8086 и i80286, работающими в реальном режиме. При этом архитектура i80386 была почти идентична архитектуре 86-го процессора, для программиста же он вообще представлялся как ЦП i8086, выполняющий соответствующие программы с большей скоростью и обладающий расширенной системой команд и регистрами. Благодаря этим качествам 32-разрядного продукта компания сохранила прежних клиентов, которые хотели модернизировать свои системы, не отказываясь от имевшегося задела в области программного обеспечения, и привлекла тех, кому изначально требовалась высокая скорость обработки информации. Этот "двунаправленный" маркетинговый ход затронул как разработчиков и пользователей ПК, так и специалистов, занимающихся сложными системами управления. Одно из основных ограничений реального режима было связано с предельным объемом адресуемой памяти, равным 1 Мбайт. От него свободен защищенный режим, позволяющий воспользоваться всеми преимуществами архитектуры нового ЦП. Размер адресного пространства в этом случае увеличивался до 4 Гбайт, а объем поддерживаемых программ - до

64 Тбайт. Производителям ПО это позволяло задействовать достаточно гибкие методы разработки и создавать более крупные программные пакеты.

Для конечных пользователей выполнение приложений, рассчитанных на работу в реальном и защищенном режимах, происходило без каких-либо функциональных отличий, поскольку управление обоими режимами базировалось на средствах ОС и специальном прикладном ПО. Однако системы защищенного режима обладали более высоким быстродействием и возможностями организации истинной многозадачности. Наконец, режим виртуального МП открывал возможность одновременного исполнения ОС и прикладных программ, написанных для МП i8086, i80286 и i80386. Поскольку объем памяти, адресуемой 386-м процессором, не ограничен значением 1 Мбайт, он позволял формировать несколько виртуальных сред i8086. Немаловажно, что эти среды могли порождаться в одно и то же время, а механизм защищенного режима обеспечивал ОС и ее прикладным задачам использование различных областей памяти. Благодаря таким возможностям аппаратуры, можно было вместо нескольких ЦП типа i8086 использовать один процессор i80386, сохранив львиную долю имевшегося ПО.

Примерно в этот же период IBM и Microsoft приступили к разработке новой многозадачной ОС с графическим интерфейсом пользователя. Архитектура кристалла, явно ориентированная на многозадачные среды, заставляет предположить участие в формировании технических требований к первому 32-разрядному МП не только IBM (о чем уже говорилось), но и Microsoft. И хотя впоследствии пути этих компаний разошлись (достаточно упомянуть противостояние OS/2 и Windows), процессор одинаково успешно работал с первыми версиями обеих систем. Стремление удовлетворить запросы потребителей всех категорий привело руководство Intel к идее повторить свой удачный опыт с процессором i8088, выпустив вариант 386-го МП с 16-разрядной внешней шиной данных (при сохранении внутренней 32-разрядной архитектуры). Существующий прибор получил обозначение i80386SX и был анонсирован 16 июня 1988 г., а уже менее чем через полгода пользователям были предложены первые ПК на его основе. Поскольку

эти модели стоили дешевле компьютеров с ЦП 80386DX, многие потребители вполне справедливо рассматривали их как начальную ступень в применении вычислительной техники.

Так Intel создала еще один сектор рынка. В конце 80-х годов степень интеграции микросхем быстро приближалась к 1млн транзисторов на кристалле, и некоторые специалисты шутили, что еще неизвестно, что же делать с такой БИС, а журнал "Electronics" в одном из традиционных новогодних поздравлений пожелал Гордону Муру вдохновения, чтобы придумать ей достойное применение.

Вдохновение не оставило президента Intel, и 10 апреля 1989 г. корпорация объявила о начале выпуска 32-разрядного прибора второго поколения - i80486, ставшего после устройств i8080 и i8086 еще одним долгожителем. Архитектура нового ЦП (рис. 5) отчасти напоминала строение своего предшественника, но вместе с тем имела и ряд коренных отличий. 1,2 млн транзисторов позволили разработчикам реализовать на кристалле не только кэш-память, но и математический сопроцессор. Такое техническое решение свело к возможному минимуму число чипов на плате и самым благоприятным образом сказалось на стоимости готовых систем. В отличие от имевшегося в МП 80386 двухвходового множественного ассоциативного кэша в i80486 использовался более эффективный четырехвходовой буфер, который, будучи размещенным в чипе, мог работать на тактовой частоте процессора. Как и предыдущий МП, новый прибор функционировал в трех режимах и был ориентирован на многозадачные среды. За счет интеграции математического сопроцессора в БИС, а также модернизации его архитектуры производительность на задачах вычислительного характера возросла в 3 - 4 раза. Общая же производительность 486-го превышала аналогичный параметр своего предшественника в 4 - 5 раз.

Блестящие характеристики процессора i80486 позволили ему быстро завоевать симпатии производителей и пользователей, и через два года модели ПК на его базе полностью вытеснили изделия предыдущего поколения. В течение шести лет кристалл безраздельно господствовал на рынке. Институт "вторых поставщиков", возникший еще в эпоху 8-разрядных систем, продолжал су-

ществовать, и изготовлением микросхем класса 486 занимались такие гиганты электронной промышленности, как IBM, Texas Instruments и AMD. На совершенствовании архитектуры кристаллов Intel родились и сделали имя Cuth и NexGen. Конкуренты оказались достаточно шустрыми и за ними требовался присмотр. Поэтому, даже приступив к проработкам процессора следующего поколения, компания не оставляла свое последнее творение без внимания.

Ровно через два года после выпуска i80486 появилась упрощенная версия кристалла (без сопроцессора), получившая по далеко не очевидной аналогии с предшествующей моделью обозначение i80486SX. Дальнейшее совершенствование пошло по пути увеличения тактовой частоты: были представлены версии на 50, 66, 75 и 100 МГц. Разработку вариантов с более высокими тактовыми частотами сочли нецелесообразной, и компании-конкуренты получили в этом отношении карт-бланш. Intel же была целиком устремлена в будущее.

Pentium - пятое поколение микропроцессоров. Стремительное усложнение программного обеспечения и постоянное расширение сферы применения компьютеров настоятельно требовали существенного роста вычислительной мощности центральных процессоров ПК. Ко всему прочему на пятки стали наступать и RISC-процессоры. Хотя в конце 80-х годов некоторые эксперты предсказывали близкий конец кристаллов CISC, корпорация Intel вполне справедливо считала, что до этого еще далеко и в микропроцессорах использованы не все возможности CISC-архитектуры. Кроме того, фирме вряд ли простили бы отказ от программной совместимости с предшествующими моделями - стоимость накопленного системного и прикладного ПО уже измерялась в миллиардах долларов.

Как это случалось не раз, проработки нового процессора начались, когда проект создания 486-го МП вступил в заключительную стадию (начало 1989 г.). Менеджером команды "архитекторов" кристалла следующего поколения был назначен Дональд Альперт (Donald Alpert) из отделения микропроцессоров корпорации Intel, а работы по проектированию арифметического устройства с плавающей точкой возглавил Дрор Эвنون (Dror Avnon).

В основу продукта была положена суперскалярная архитектура (еще один атрибут из мира мэйнфреймов), которая и дала возможность получить пятикратное повышение производительности по сравнению с моделью 486DX.

Высокая скорость выполнения команд достигалась благодаря двум 5-ступенчатым конвейерам, позволявшим одновременно исполнять несколько инструкций. Для постоянной загрузки обоих конвейеров из кэша требуется широкая полоса пропускания. Совмещенный буфер команд и данных обеспечить ее не мог, и разработчики воспользовались решением из арсенала RISC-процессоров, оснатив Pentium отдельными буферами команд и данных. При этом обмен информацией с памятью через кэш данных осуществлялся совершенно независимо от процессорного ядра, а буфер инструкций был связан с ним через высокоскоростную 256-разрядную внутреннюю шину. Несмотря на то что новый кристалл был спроектирован как 32-разрядный, для связи с остальными компонентами системы использовалась внешняя 64-разрядная шина данных с максимальной пропускной способностью 528 Мбайт/с. Еще одной "изюминкой" архитектуры, позаимствованной у представителей универсальных ЭВМ стала схема предсказания переходов.

По оценкам специалистов компании, это дополнительно увеличило производительность ЦП на 20 - 25%. Обработка графической информации, мультимедиа-приложения, а также все более интенсивное использование ПК для решения задач вычислительного характера требуют высокой производительности при выполнении операций с плавающей точкой. Скоростные характеристики этого блока получены за счет применения конвейерной архитектуры устройства (8-ступенчатый конвейер FPU позволил выдавать результаты каждый такт), а также благодаря аппаратной реализации (вместо традиционной микропрограммной) основных арифметических операций. Сложение, умножение и деление выполнялись отдельными высокопроизводительными блоками умножителя, сложителя и делителя. Средства для работы с плавающей точкой поддерживали стандарт IEEE 754 и операции с одинарной, двойной и расширенной (80-битовый формат) точностью, а также набор базовых трансцендентных функций. В итоге по скорости вы-

полнения команд с плавающей точкой Pentium в пять - семь раз превзошел процессор 486DX2/50 и почти на порядок - микросхему 486DX/33. Выход новинки на рынок был обставлен корпорацией в ее лучших традициях (вспомним акцию Operation Crush), начиная от мощной рекламной кампании и кончая присвоением процессору собственного имени.

Предварительная информация о характеристиках этого ЦП, которая, кстати, незначительно разошлась с реальными параметрами прибора, подогрела к нему интерес как поставщиков компьютеров, так и пользователей. Пристальное внимание Intel традиционно уделила тесному взаимодействию с разработчиками ПО, которые получили спецификации процессора Pentium одними из первых и очередные версии приложений проектировали уже с учетом возможностей кристалла. Таким образом, процессор вышел в свет в сопровождении почетного эскорта новых программных продуктов.

Несмотря на ряд проблем, практически всегда возникающих при выпуске новых изделий, Pentium завоевал популярность и начал быстро теснить на рынке своего предшественника. Этому способствовала и политика самой компании, которая, стремясь обеспечить ускоренный переход потребителей на Pentium, вскоре прекратила работы по совершенствованию микросхем i80486 и даже начала сворачивать их производство, что, кстати, вызвало нескрываемое раздражение некоторых крупных потребителей продукции Intel, в частности Compaq. Ко всему прочему, теперь Intel уже никому не хотела продавать свое ноу-хау и институт вторых поставщиков приказал долго жить. С одной стороны, необходимости в утверждении архитектуры, и без того давно ставшей мировым стандартом, более не было, а производственные мощности корпорации стали вполне достаточны для удовлетворения спроса на ее изделия; с другой - создание Pentium оказалось настолько дорогим удовольствием (по разным оценкам, Intel вложила в этот проект 4 - 6 млрд дол.), что отдавать даже малую часть рынка становилось довольно рискованным. Хотя объемы продаж систем на базе нового ЦП росли буквально не по дням, а по часам (в 1994 г. их было продано 5,5 млн, а в 1995 г. - уже 18,4 млн), почивать на

лаврах компания не собиралась и постоянно улучшала рабочие характеристики чипов за счет увеличения тактовой частоты.

Начав в 1993 г. с отметки в 60 МГц и быстро миновав промежуточные барьеры в 66; 75; 90; 100; 120; 133 и 166 МГц, в этом году Intel довела тактовую частоту Pentium II - представитель второго поколения семейства P6. Помимо того что он дополнен MMX-командами, этот кристалл содержит в два раза большую кэш-память по сравнению с Pentium Pro - по 16 Кбайт (вместо восьми) для команд и данных - и взаимодействует с кэшем L2 не на свой полный тактовой, а на вдвое меньшей частоте.

Pentium II размещается не в двухотсекном керамическом корпусе вместе со специализированным кэшем L2, а представляет собой традиционную однокристалльную конструкцию, в которой предусмотрены отдельные соединения с системной шиной и интерфейс кэша L2. Однако Intel не продает этот кристалл сам по себе; процессор устанавливается на небольшой печатной плате вместе с кэшем L2, построенном на кристаллах памяти SRAM. Затем эта сборка, которую Intel называет картриджем SEC (Single Edge Contact - с односторонним расположением контактов), помещается в корпус из пластмассы и металла. Именно эта конструкция Intel носит название Pentium II.

Интерфейс кэша L2 процессора Pentium II имеет в два раза меньшую скорость, чем у кэша Pentium Pro, что компенсируется до некоторой степени большей емкостью кэша L1 процессора. Гораздо важнее то, что благодаря двукратному уменьшению скорости интерфейса кэша становится значительно проще увеличить тактовую частоту ЦП; в настоящее время выпускаются 233-, 266- и 300-МГц модели Pentium II, а в будущем году намечается достичь отметки 400 МГц.

Применение для кэша Pentium II микросхем памяти, выпускаемых другими изготовителями, высвобождает производственные мощности фирмы Intel, а благодаря острой конкуренции на рынке SRAM сохраняется низкий уровень себестоимости. Это позволило Intel сделать 512-Кбайт кэш L2 стандартной принадлежностью Pentium II; вариант 512-Кбайт кэша для Pentium Pro обходится дорого и используется очень редко. В результате хотя, ин-

терфейс кэша L2 процессора Pentium II работает с меньшей скоростью, емкости его кэшей L1 и L2 в два раза больше.

Одно из ограничений, присущих нынешней архитектуре Pentium II, состоит в том, что она позволяет строить только двухпроцессорные системы, в то время как Pentium Pro может работать в четырехпроцессорной конфигурации. По этой причине до конца нынешнего года многие разработчики серверов будут по-прежнему ориентироваться на Pentium Pro. Даже в некоторых одно- и двухпроцессорных серверах будет применяться Pentium Pro, поскольку для выполнения прикладных задач, требующих обработки больших объемов данных, преимущества высокоскоростного кэша L2 могут оказаться гораздо важнее, нежели большая емкость кэшей и высокая тактовая частота. Кроме того, серверы, как правило, мало что выигрывают от технологии MMX. Однако в 1998 г. появится новый вариант Pentium II (описанный ниже в этой статье), который вытеснит Pentium Pro с рынка серверов.

Приложение 3. Сводные данные о МП Intel

Процессор Intel "Xeon"

Процессор Intel Xeon (2; 1,7; 1,5 и 1,4 ГГц)

Дата выпуска: 25 сентября 2001 г. (2 ГГц)

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (интегрированная)

Тип корпуса: Organic Lan Grid Array 603 (OLGA 603)

Частота системной шины: 400 МГц

Потоковые SIMD расширения SSE2

Область применения: Двухпроцессорные серверы и рабочие станции

Процессор Intel "Pentium" 4

Процессор Intel Pentium 4 произведенный по 0.18-микронной технологии (2; 1,9; 1,8; 1,7; 1,6; 1,5 и 1,4 ГГц)

Дата выпуска: 27 августа 2001 г. (2, 1.9 ГГц); 2 июля 2001 г. (1.8, 1.6 ГГц); 23 апреля 2001 г. (1.7 ГГц); 20 ноября 2000 г. (1.5, 1.4 ГГц)

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (интегрированная)

Тип корпуса: PGA423, PGA478

Частота системной шины: 400 МГц

Потоковые SIMD расширения SSE2

Область применения: офисные и домашние ПК, одно- и двухпроцессорные серверы и рабочие станции

Процессор Pentium III Xeon

Процессор Pentium III Xeon, произведенный по 0.18-микронной технологии (900 МГц)

Дата выпуска: 21 марта 2001 г. (900 МГц)

Кэш-память второго уровня: 2 МБ Advanced Transfer Cache (интегрированная)

Тип корпуса: SC330

Частота системной шины: 100 МГц

Разрядность системной шины: 64 разрядов

Адресуемая память: 64 ГБ

Область применения: сервера конечного уровня, четырех-восьмипроцессорные рабочие станции

Процессор Pentium III Xeon, произведенный по 0.18-микронной технологии (933 МГц)

Дата выпуска: 24 мая 2000 г. (933 МГц)

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (интегрированная)

Тип корпуса: SC330

Частота системной шины: 133 МГц

Разрядность системной шины: 64 разрядов

Адресуемая память: 64 ГБ

Область применения: офисные и домашние ПК, одно- и двухпроцессорные серверы и рабочие станции

Процессор Pentium III Xeon, произведенный по 0.18-микронной технологии (700 МГц)

Дата выпуска: 21 марта 2001 г. (700 МГц)

Кэш-память второго уровня: 1 и 2 МБ Advanced Transfer Cache (интегрированная)

Тип корпуса: SC330

Частота системной шины: 100 МГц

Разрядность системной шины: 64 разрядов

Адресуемая память: 64 ГБ

Область применения: четырех-восьмипроцессорные рабочие станции

Процессор Pentium III Xeon, произведенный по 0.18-микронной технологии (866, 800, 733, 667 и 600 МГц)

Дата выпуска: 25 октября 1999 г. (733, 667 и 600 МГц); 12 января 2001 г. (800 МГц); 10 апреля 2000 г. (866 МГц)

Число транзисторов: 28 миллионов (0.18-микронная технология)

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (интегрированная)

Тип корпуса: Single Edge Contact Cartridge (S.E.C.C. 2)

Частота системной шины: 133 МГц

Разрядность системной шины: 64 разрядов

Адресуемая память: 64 ГБ

Область применения: двухпроцессорные рабочие станции

Процессор Pentium III Xeon, произведенный по 0.18-микронной технологии (500 и 550 МГц)

Дата выпуска: 17 марта 1999 г.

Тактовая частота: 500 и 550 МГц

Число транзисторов: 9,5 миллионов (0.25-микронная технология)

Кэш-память второго уровня: 512 КБ и 1-2 МБ Advanced Transfer Cache (интегрированная)

Тип корпуса: Single Edge Contact Cartridge (S.E.C.C. 2)

Частота системной шины: 100 МГц

Разрядность системной шины: 64 разрядов

Адресуемая память: 64 ГБ

Область применения: потребительские ПК, одно-и двухпроцессорные серверы и рабочие станции

Процессор Intel Pentium III с технологией сверхнизкого энергопотребления

Дата выпуска: 13 ноября 2001 г.

Тактовая частота: 700 МГц

Частота системной шины: 100 МГц

Технология производства: 0.13 микрон
Кэш-память второго уровня: 512 КБ
Тип корпуса: uFCBGA
SIMD Extensions
Наряжение ядра: 1.1 В
Область применения: рабочие станции

Процессор Intel Pentium III для мобильных ПК

Дата выпуска: 30 июля 2001 г.
Тактовая частота: 1,13; 1,06; 1 ГГц
Частота системной шины: 133 МГц
Технология производства: 0.13 микрон
Кэш-память второго уровня: 512 КБ
Тип корпуса: Micro FCBGA/PGA
SIMD Extensions
Наряжение ядра: 1.4 В в режиме максимальной производительности и 1.15 в режиме оптимальной производительности
Напряжение питания: 2 В в режиме оптимальной производительности
Область применения: полный спектр мобильных ПК

Процессор Intel Pentium III для мобильных ПК

Дата выпуска: 30 июля 2001 г.
Тактовая частота: 933, 866 МГц
Частота системной шины: 133 МГц
Технология производства: 0.13 микрон
Кэш-память второго уровня: 512 КБ
Тип корпуса: Micro FCBGA/PGA
SIMD Extensions
Наряжение ядра: 1.5 В в режиме максимальной производительности и 1.05 в режиме оптимальной производительности
Напряжение питания: 1 В в режиме оптимальной производительности
Область применения: полный спектр мобильных ПК

Процессор Pentium III со сверхнизким энергопотреблением и технологией Intel SpeedStep \geq (600, 500 МГц) для мобильных

ПК

Дата выпуска: 21 мая 2001 г. (600 МГц); 30 января 2001 г. (500 МГц)

Тактовая частота: 600, 500, 300(режим оптимальной производительности) МГц

Технология производства: 0.13 микрон

Кэш-память: 512 КБ (интегрированная)

Тип корпуса: Ball Grid Array (BGA)

Частота системной шины: 100 МГц

Наряжение ядра: 1.1 В при 600, 500 МГц; 1 В в режиме оптимальной производительности

Напряжение питания: < 1 В при 600, 500 МГц; < 0.5 при 300 МГц

Область применения: офисные и пользовательские мобильные ПК

Технология низкого энергопотребления**Процессор Pentium III с низким энергопотреблением и технологией Intel SpeedStep \geq (750, 700, 600 МГц) для мобильных ПК**

Дата выпуска: 21 мая 2001 г. (750 МГц); 27 февраля 2001 г. (700 МГц); 19 июня 2000 г. (600 МГц)

Технология производства: 0.18 микрон

Кэш-память второго уровня: 512 КБ (интегрированная)

Тип корпуса: Ball Grid Array (BGA)

Частота системной шины: 100 МГц

Потоковые SIMD расширения

Наряжение ядра: 1.1 В в режиме оптимальной производительности

Напряжение питания: 1 В в режиме оптимальной производительности

Область применения: офисные и пользовательские мобильные ПК

Процессоры для мобильных ПК**Процессор Pentium III с технологией Intel SpeedStep \geq (1 ГГц, 900, 850, 800, 750 МГц)**

Дата выпуска: 19 марта 2001 г. (1 ГГц, 900 МГц); 25 сентября 2000 г. (850, 800 МГц); 19 июня 2000 (750 МГц)

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (ин-

тегрированная)

Тип корпуса: Mobile Module, Ball Grid Array (BGA) или Micro Pin Grid Array (micro PGA)

Частота системной шины: 100 МГц

Адресуемая память: 64 ГБ

Напряжение ядра: 1.6 В

Напряжение питания: <2 В

Область применения: офисные и пользовательские ПК

Процессор Pentium III с технологией Intel SpeedStep \geq (700, 650, 600 МГц)

Дата выпуска: 18 января 2000 г. (650, 600 МГц); 24 апреля 2000 г. (700 МГц); 19 июня 2000 г. (750 МГц)

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (интегрированная)

Тип корпуса: Mobile Module, Ball Grid Array (BGA) или Micro Pin Grid Array (micro PGA)

Частота системной шины: 100 МГц

Адресуемая память: 64 ГБ

Напряжение ядра: 1.6 В

Область применения: офисные и пользовательские ПК

Процессор Pentium III (500, 450, 400 МГц), произведенный по технологии 0.18-микрон

Дата выпуска: 25 октября 1999 г. (500, 450, 400 МГц)

Число транзисторов: 28 миллионов (0.18-микронная технология)

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (интегрированная)

Тип корпуса: Mobile Module, Ball Grid Array (BGA) или Micro Pin Grid Array (micro PGA)

Частота системной шины: 100 МГц

Разрядность системной шины: 64 разрядов

Адресуемая память: 64 ГБ

Напряжение ядра: 1.6 В для 500 и 450 МГц; 1.35 В для 400 МГц

Область применения: пользовательские ПК и двухпроцессорные сервера

Для встраиваемых применений

Процессор Pentium III (700 МГц) для встраиваемых применений

Дата выпуска: 19 марта 2001 г.

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (интегрированная)

Технология производства: 0.18 микрон

Тип корпуса: Ball Grid Array (BGA)

Напряжение питания: <2 В

Частота системной шины: 100 МГц

Потоковые SIMD расширения

Адресуемая память: 64 ГБ

Напряжение ядра: 1.35 В

Напряжение питания:

Настольные ПК

Процессор Pentium III (933 МГц), произведенный по технологии 0.18-микрон

Дата выпуска: 24 мая 2000 г. (933 МГц)

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (интегрированная)

Тип корпуса: Single Edge Contact Cartridge (S.E.C.C. 2), Flip-Chip Pin Grid Array (FC-PGA)

Частота системной шины: 133 МГц

Разрядность системной шины: 64 разрядов

Адресуемая память: 64 ГБ

Область применения: офисные и пользовательские ПК и двухпроцессорные сервера

Процессор Pentium III (1 ГГц, 933, 866/850 МГц), произведенный по технологии 0.18-микрон

Дата выпуска: 8 марта 2000 г. (1 ГГц); 20 марта 2000 г. (866, 850 МГц); 24 мая 2000 г. (933 МГц)

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (интегрированная)

Тип корпуса: Single Edge Contact Cartridge (S.E.C.C. 2), Flip-Chip Pin Grid Array (FC-PGA)

Частота системной шины: 100 и 133 МГц

Разрядность системной шины: 64 разрядов

Адресуемая память: 64 ГБ

Область применения: офисные и пользовательские ПК и двухпроцессорные сервера

Процессор Pentium III (500, 533, 550, 600, 650, 667, 700 и 733 МГц), произведенный по технологии 0.18-микрон

Дата выпуска: 25 октября 1999 г. (500, 533, 550, 600, 650, 667, 700 и 733 МГц)

Число транзисторов: 28 миллионов (0.18-микронная технология)

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (интегрированная)

Тип корпуса: Single Edge Contact Cartridge (S.E.C.C. 2), Flip-Chip Pin Grid Array (FC-PGA)

Частота системной шины: 100 и 133 МГц

Разрядность системной шины: 64 разрядов

Адресуемая память: 64 ГБ

Область применения: офисные и пользовательские ПК и двухпроцессорные сервера

Процессор Pentium III (450, 500, 550 и 600 МГц)

Дата выпуска: 26 февраля 1999 г. (450, 500 МГц); 17 мая 1999 г. (500 МГц); 2 августа 1999 г. (600 МГц)

Тактовая частота: 450, 500, 550 и 600 МГц

Число транзисторов: 9,5 миллионов (0.25-микронная технология)

Кэш-память второго уровня: 256 КБ

Тип корпуса: Single Edge Contact Cartridge (S.E.C.C. 2)

Частота системной шины: 100 МГц

Разрядность системной шины: 64 разрядов

Адресуемая память: 64 ГБ

Область применения: офисные и пользовательские ПК и двухпроцессорные сервера

Процессор Pentium II Xeon(450 МГц)

Объявлен: 5 января 1999 г.

Тактовая частота: 450 МГц

Данные по производительности: серверы, рабочие станции

Кэш L2: 512 КБ, 1 МБ и 2 МБ

Количество транзисторов: 7.5 млн

Тип корпуса процессора: Картридж с односторонним контактом (S.E.C)

Частота шины: 100 МГц

Ширина полосы пропускания шины: 8 байт

Адресуемая память: 64 Гигабайт

Виртуальная память: 64 Терабайт

Габариты модуля: высота 4.8" x ширина 6.0" x глубина .73"

Применение: 4-процессорные серверы и рабочие станции

Процессор Pentium II Xeon (450 МГц)

Объявлен: 6 октября 1998 г.

Тактовая частота: 450 МГц

Данные по производительности: рабочие станции, серверы

Кэш L2: 512 КБ

Количество транзисторов: 7.5 млн

Тип корпуса: Картридж с односторонним контактом (S.E.C)

Частота шины: 100 МГц

Ширина полосы пропускания шины: 8 bytes

Адресуемая память: 64 Гигабайт

Виртуальная память: 64 Терабайт

Габариты модуля: высота 4.8" x ширина 6.0" x глубина .73"

Применение: Двухпроцессорные рабочие станции и серверы

Процессор Pentium II Xeon (400 МГц)

Объявлен: 29 июня 1998 г.

Тактовая частота: 400 МГц

Данные по производительности: рабочие станции, серверы

Кэш L2: 512 КБ и 1 МБ

Количество транзисторов: 7.5 млн

Тип корпуса: Картридж с односторонним контактом (S.E.C)

Частота шины: 100 МГц

Ширина полосы пропускания шины: 8 bytes

Адресуемая память: 64 Гигабайт
Виртуальная память: 64 Терабайт
Габариты модуля: высота 4.8" x ширина 6.0" x глубина .73"
Применение: Серверы и рабочие станции среднего уровня и выше

Процессор Pentium II

Процессор Pentium II для мобильных ПК (400 МГц)

Объявление: 14 июня 1999 г.

Тактовая частота: 400 МГц; данные по производительности

Количество транзисторов: 27.4 млн (0.18-мкм процесс), 256 КБ встроенной кэш L2

Тип корпуса: Ball Grid Array (BGA): число выводов = 615

Габариты: ширина = 31 мм; длина = 35 мм

Напряжение ядра: 1.5 В

Питание зависит от частоты: 400 МГц = 7.5 Вт

Применение: мобильные ПК

Процессор Pentium II для мобильных ПК (400 МГц)

Объявление: 14 июня 1999 г.

Тактовая частота: 400 МГц; данные по производительности

Количество транзисторов: 27.4 млн (0.25-мкм процесс), 256 КБ встроенной кэш L2

Тип корпуса: Mini-Cartridge and MMC1, MMC2

Напряжение ядра: 1.5 В

Питание зависит от частоты: 400 МГц = 7.5 Вт

Применение: мобильные ПК

Процессор Pentium II для мобильных ПК (266, 300, 333 и 366 МГц)

Объявление: 25 января 1999 г.

Тактовые частоты: 266, 300, 333 и 366 МГц; данные по производительности

Количество транзисторов: 27.4 млн (0.25-мкм процесс), 256 КБ встроенной кэш L2

Тип корпуса: Ball Grid Array (BGA): число выводов = 615

Габариты: ширина = 31 мм; длина = 35 мм

Напряжение ядра: 1.6 В

Питание зависит от частоты: 366 МГц = 9.5 Вт; 333 МГц = 8.6 Вт; 300 МГц = 7.7 Вт; 266 МГц = 7.0 Вт

Применение: мобильные ПК

Процессор Pentium II для мобильных ПК (300 МГц)

Объявлен: 9 сентября 1998 г.

Тактовая частота: 300 МГц; данные по производительности

Количество транзисторов: 7.5 млн (0.25-мкм процесс), 512 К кэш L2

Разрядность шины: 64-bit

Адресуемая память: ~68 Гигабайт

Мобильный мини-картридж: 240 выводов

Мобильный модуль: 280(mmc1)/400(mmc2) выводов

Напряжение ядра: 1.6 В

Мощность: 9.0 Вт (включая ядро CPU и 512 КБ кэш L2)

Применение: мобильные ПК

Процессор Pentium II для мобильных ПК (233 и 266 МГц)

Объявлен: 2 апреля 1998 г.

Тактовая частота: 233, 266 МГц; данные по производительности

Количество транзисторов: 7.5 млн (0.25-мкм процесс), 512 К кэш L2

Разрядность шины: 64-bit

Адресуемая память: ~68 Гигабайт

Мобильный мини-картридж: 240 выводов

Мобильный модуль: 280(mmc1)/400(mmc2) выводов

Напряжение ядра: 1.7 В

Мощность: 7.5 Вт для 233 МГц и 8.6 Вт для 266 МГц (включая ядро CPU и 512 КБ кэш L2)

Применение: мобильные ПК

Настольные ПК

Процессор Pentium II (450 МГц)

Объявлен: 24 августа 1998 г.

Тактовая частота: 450 МГц;

Количество транзисторов: 7.5 млн (0.25-мкм процесс)

Картридж с односторонним контактом (S.E.C), 242 вывода
Габариты модуля: 5.505" x 2.473" x 0.647"

Частота шины: 100 МГц

Разрядность шины: 64-разрядная системная шина

Адресуемая память: 64 Гигабайт

Применение: Бизнес- и потребительские ПК, одно- и двухпроцессорные серверы и рабочие станции

Процессор Pentium II (350 и 400 МГц)

Объявлен: 15 апреля 1998 г.

Тактовая частота: 350, 400 МГц;

Количество транзисторов: 7.5 млн (0.25-мкм процесс), кэш L2 512К

Картридж с односторонним контактом (S.E.C), 242 вывода

Габариты модуля: 5.505" (12.82cm) x 2.473" (6.28cm) x 0.647" (1.64cm)

Частота шины: 100 МГц

Применение: ПК для бизнеса и для широкого круга потребителей, одно- и двухпроцессорные серверы и рабочие станции

Процессор Pentium II (333 МГц)

Объявлен: 26 января 1998 г.

Тактовая частота: 333 МГц (12.8 SPECint95, 9.14 SPECfp95, 8.32 SPECfpbase)

Количество транзисторов: 7,5 млн (0,25-мкм технология), кэш-память второго уровня 512 КБ

Разрядность шины: 64-битная системная шина с аппаратной коррекцией ошибок (ECC); 64-битная шина кэш с оптимизацией ECC

Адресуемая память: 64 гигабайт

Виртуальная память: 64 терабайт

Корпус с односторонним контактом (Single Edge Contact Cartridge - S.E.C), 242 вывода

Габариты модуля: 12,82 см x 6,28 см x 1,64 см

Частота шины: 66 МГц

Применение: ПК для бизнеса и для широкого круга потребителей, одно- и двухпроцессорные серверы и рабочие станции

Процессор Pentium II (300, 266, 233 МГц)

Объявлен: 7 мая 1997 г.

Тактовая частота: 300, 266, 233 МГц (11.7 SPECint95, 8.15 SPECfp95)

Количество транзисторов: 7,5 млн (0,35-мкм технология), кэш-память второго уровня 512 Кб

Ширина полосы пропускания шины: 64-битная системная шина с ЕСС; 64-битная шина кэш с оптимизацией ЕСС

Адресуемая память: 64 Гигабайт

Виртуальная память: 64 Терабайт

Корпус с односторонним контактом (S.E.C.), 242 вывода

Габариты модуля: 12,82 см x 6,28 см x 1,64 см

Применение: настольные компьютеры высшего уровня для бизнеса, рабочие станции и серверы

Процессор Intel Celeron

Технология сверхнизкого энергопотребления

Мобильный процессор Celeron со сверхнизким энергопотреблением и технологией Intel SpeedStep \geq (600 и 500 МГц)

Объявлены: 21 мая 2001 г. (600 МГц); 30 января 2001 г.

Тактовые частоты: 500 и 600 МГц; **данные по производительности**

Кэш-память второго уровня: 128 КБ

Тип корпуса: Ball Grid Array (BGA)

Частота шины: 100 МГц

Напряжение ядра: 1.1 В

Питание: <1 Вт

Применение: офисные и потребительские мобильные ПК

Мобильный процессор Celeron с низким энергопотреблением и технологией Intel SpeedStep \geq (600 МГц)

Объявлены: 21 мая 2001 г. (600 МГц); 30 января 2001 г.

Тактовые частоты: 600 МГц; **данные по производительности**

Кэш-память: 256 КБ

Технология производства: 0.18 микрон

Тип корпуса: Ball Grid Array (BGA)

Частота шины: 100 МГц

Напряжение ядра: 1.35 В

Питание: <2 Вт

Применение: ультра-портативные мобильные ПК

Процессоры для мобильных ПК

Мобильный процессор Intel Celeron (850, 800, 750, 700, 650, 550, 500, 450 МГц)

Объявлены: 2 июля 2001 г. (850 МГц); 21 мая 2001 г. (800 МГц); 19 марта 2001 г. (750 МГц); 25 сентября 2000 г. (700 МГц); 19 июня 2000 г. (650, 600 МГц); 24 апреля 2000 г. (550 МГц); 14 февраля 2000 г. (500, 450 МГц)

Тактовые частоты: 450-850 МГц; данные по производительности

Технология производства: 0.18 микрон; 128 КВ кэш-памяти второго уровня

Потоковые SIMD расширения Тип корпуса: Ball Grid Array (BGA2) and Pin Grid Array(PGA2)

Габариты: ширина = 31 мм; длина = 35 мм

Напряжение ядра: 1.6 В

Применение: недорогие мобильные ПК

Мобильный процессор Intel Celeron (466, 433 МГц)

Объявлены: 15 сентября 1999 г.

Тактовые частоты: 466, 433 МГц; данные по производительности

Количество транзисторов: 18,9 млн (0,25-мкм технология), кэш-память второго уровня 128 Кб

Тип корпуса: Ball Grid Array (BGA), 615 выводов

Габариты: ширина = 31 мм; длина = 35 мм

Напряжение ядра: 1.9 В

Питание: 466 МГц = 15.6 Вт; 433 МГц = 14.5 Вт

Применение: недорогие мобильные ПК

Мобильный процессор Intel Celeron (400 МГц)

Объявлен: 14 июня 1999 г.

Тактовая частота: 400 МГц; данные по производительности

Количество транзисторов: 18,9 млн (0,25-мкм технология), кэш-память второго уровня 128 Кб

Тип корпуса: Ball Grid Array (BGA), 615 выводов

Габариты: ширина = 31 мм; длина = 35 мм

Напряжение ядра: 1.6 В

Питание: 400 МГц = 9.2 Вт

Применение: недорогие мобильные ПК

Мобильный процессор Intel Celeron (366 МГц)

Объявлен: 17 мая 1999 г.

Тактовая частота: 366 МГц; данные по производительности

Количество транзисторов: 18,9 млн (0,25-мкм технология), кэш-память второго уровня 128 Кб

Тип корпуса: Ball Grid Array (BGA), 615 выводов

Габариты: ширина = 32 мм; длина = 37 мм

Напряжение ядра: 1.6 В

Питание: 9.0 Вт

Применение: недорогие мобильные ПК

Мобильный процессор Intel Celeron (333 МГц)

Объявлен: 5 апреля 1999 г.

Тактовая частота: 333 МГц; данные по производительности

Количество транзисторов: 18,9 млн (0,25-мкм технология), кэш-память второго уровня 128 Кб

Тип корпуса: Ball Grid Array (BGA), 615 выводов

Габариты: ширина = 31 мм; длина = 35 мм

Напряжение ядра: 1.6 В

Питание: 333 МГц = 8.6 Вт

Применение: недорогие мобильные ПК

Мобильный процессор Intel Celeron (266 и 300 МГц)

Объявлен: 25 января 1999 г.

Тактовые частоты: 266 и 300 МГц; данные по производительности

Количество транзисторов: 18.9 млн (0.25-мкм процесс), 128К встроенной кэш L2

Тип корпуса: Ball Grid Array (BGA), число выводов = 615

Габариты: ширина = 31 мм; длина = 35 мм
Напряжение ядра: 1.6 В
Питание: 300 МГц = 7.7 Вт; 266 МГц = 7.0 Вт
Применение: недорогие мобильные ПК

Настольные ПК

Процессор Intel Celeron (1,2 ГГц)

Объявлен: 2 октября 2001 г.
Тактовая частота: 1.2 ГГц
Частота шины: 100 МГц
Технология производства: 0.13 микрон Кэш-память второго уровня: 128 КВ
Тип корпуса: FCPGA
SIMD Extensions Применение: недорогие ПК

Процессор Intel Celeron (1,1; 1,0 ГГц, 950, 900, 850, 800 МГц)

Объявлены: 13 ноября 2000 г. (1.1, 1.0 ГГц, 950 МГц); 2 июля 2001 г. (900 МГц); 21 мая 2001 г. (850 МГц);
Тактовые частоты: 800 МГц - 1.1 ГГц данные по производительности
Кэш-память второго уровня: 128 КВ
Тип корпуса: Flip-Chip Pin Grid Array (FC-PGA)
Частота шины: 100 МГц
Адресуемая память: 4 Гигабайт
Применение: недорогие ПК

Процессор Intel Celeron (766, 733, 700, 667, 633, 600, 566 МГц)

Объявлены: 13 ноября 2000 г. (766, 733 МГц); 26 июня 2000 г. (700, 667, 633 МГц); 29 марта 2000 г. (600, 566 МГц);
Тактовые частоты: 766, 733, 700, 667, 633, 600, 566 МГц данные по производительности
Кэш-память: 128 КВ
Тип корпуса: Flip-Chip Pin Grid Array (FC-PGA)
Частота шины: 66 МГц
Адресуемая память: 4 Гигабайт
Применение: недорогие ПК

Процессор Intel Celeron (533 МГц)

Объявлена: 4 января 2000 г.

Тактовые частоты: 533 МГц данные по производительности

Количество транзисторов: 19 млн (0.25-мкм процесс)

Кэш-память: 128 КВ

Тип корпуса: Flip-Chip Pin Grid Array (PPGA), 370 выводов

Частота шины: 66 МГц

Разрядность шины: 64-бит Адресуемая память: 4 Гигабайт

Применение: недорогие ПК

Процессор Intel Celeron (466, 500 МГц)

Объявлены: 26 апреля 1999 г. (466 МГц); 2 августа 1999 г. (500 МГц)

Тактовые частоты: 466, 500 МГц данные по производительности

Количество транзисторов: 19 млн (0.25-мкм процесс)

Кэш-память: 128 КВ

Тип корпуса: Flip-Chip Pin Grid Array (PPGA), 370 выводов

Частота шины: 66 МГц

Разрядность шины: 64-бит Адресуемая память: 4 Гигабайт

Применение: недорогие ПК

Процессор Intel Celeron (433 МГц)

Объявлены: 22 марта 1999 г. (466 МГц)

Тактовые частоты: 433 МГц данные по производительности

Количество транзисторов: 19 млн (0.25-мкм процесс)

Кэш-память: 128 КВ

Тип корпуса: Single Edge Processor Package (SEPP), 242 выхода;
Plastic Pin Grid Array (PPGA), 370 выводов

Частота шины: 66 МГц

Разрядность шины: 64-бит Адресуемая память: 4 Гигабайт

Применение: недорогие ПК

Процессор Intel Celeron (400, 366 МГц)

Объявлен: 4 января 1999 г.

Тактовая частота: 400, 366 МГц; данные по производительности

Количество транзисторов: 19 млн (0.25-мкм процесс)

Корпус с односторонним контактом (SEPP), 242 вывода

Корпус Plastic Pin Grid Array (PPGA), 370 вывода

Частота шины: 66 МГц

Разрядность шины: 64-bit системная шина

Адресуемая память: 4 Гигабайт

Применение: недорогие ПК

Процессор Intel Celeron (333 МГц)

Объявлен: 24 августа 1998 г.

Тактовая частота: 333 МГц; данные по производительности

Количество транзисторов: 19 млн (0.25-мкм процесс)

Корпус с односторонним контактом (SEPP), 242 вывода

Частота шины: 66 МГц

Разрядность шины: 64-bit системная шина

Адресуемая память: 4 Гигабайт

Габариты модуля: 5" x 2.275" x .208"

Применение: базовые ПК

Процессор Intel Celeron (300А МГц)

Объявлен: 24 августа 1998 г.

Тактовая частота: 300 МГц; данные по производительности

Количество транзисторов: 19 млн (0.25-мкм процесс)

Корпус с односторонним контактом (SEPP), 242 вывода

Частота шины: 66 МГц

Разрядность шины: 64-bit системная шина

Адресуемая память: 4 Гигабайт

Габариты модуля: 5" x 2.275" x .208"

Применение: базовые ПК

Процессор Intel Celeron (300 МГц)

Объявлен: 8 июня 1998 г.

Тактовая частота: 300 МГц; данные по производительности

Количество транзисторов: 7.5 млн (0.25-мкм процесс)

Корпус с односторонним контактом (SEPP), 242 вывода

Частота шины: 66 МГц

Разрядность шины: 64-bit системная шина

Адресуемая память: 4 Гигабайт

Виртуальная память: 64 Терабайт

Габариты модуля: 5" x 2.275" x .208"

Применение: базовые ПК

Процессор Intel Celeron (266 МГц)

Объявлен: 15 апреля 1998 г.

Тактовая частота: 266 МГц; данные по производительности

Количество транзисторов: 7.5 млн (0.25-мкм процесс)

Корпус с односторонним контактом (SEPP), 242 вывода

Частота шины: 66МГц

Разрядность шины: 64-bit системная шина

Адресуемая память: 4 Гигабайт

Виртуальная память: 64 Терабайт

Габариты модуля: 5" x 2.275" x .208"

Применение: базовые ПК

Процессор Pentium Pro (200 МГц) с одним мегабайтом встроенной кэш-памяти второго уровня

Объявлен: 18 августа 1997 г.

Тактовая частота: 200 МГц (8.66 SPECint95, 6.80 SPECfp95)

Количество транзисторов: 5,5 млн (0,35-мкм технология), 1 Мб

кэш-памяти второго уровня: 62 млн (0,35-мкм технология)

Ширина полосы пропускания шины: 300-битная шина внутреннего обмена; 64-битная шина кэш-памяти второго уровня

Адресуемая память: 64 гигабайт

Виртуальная память: 64 терабайт

Число контактов: 387 (Dual Cavity Pin Grid Array Package)

Габариты модуля: 6,25 см x 6,76 см

Применение: Настольные ПК, рабочие станции и серверы высшего уровня

Процессор Pentium Pro (200, 180, 166, 150 МГц)

См. фото

Объявлен: 1 ноября 1995 г.

Тактовая частота: 200, 180, 166, 150 МГц (8.20 SPECint95, 6.21

SPECfp95 на системе Alder с 256 Кб кэш-памяти второго уровня)

Количество транзисторов: 5,5 млн (0,35-мкм технология), 256 Кб

кэш-памяти второго уровня 256 Кб: 15,5 млн (0,6-мкм технология),

512 Кб кэш-памяти второго уровня: 31 млн (0,35-мкм технология)

Ширина полосы пропускания шины: 64-битная внешняя системная шина; 64-битная шина кэш-памяти второго уровня

Адресуемая память: 64 гигабайт

Виртуальная память: 64 терабайт

Число контактов: 387 (Dual Cavity Pin Grid Array Package)

Габариты модуля: 6,25 см x 6,76 см

Применение: Настольные ПК, рабочие станции и серверы класса high-end

Процессоры для мобильных ПК

Процессор Pentium (300 МГц) с технологией MMX \geq

Объявлен: 7 января 1999 г.

Тактовая частота: 300 МГц

Количество транзисторов: 4,5 млн (0,25-мкм технология)

Ширина полосы пропускания шины: 64 бит

(Примечание: это полностью 32-разрядный микропроцессор)

Адресуемая память: ~68 Гигабайт

Тип корпуса: Tare Carrier Package (ТСР), 320 выводов

Применение: мобильные ПК и ноутбуки

Процессор Pentium (266 МГц) с технологией MMX \geq

Объявлен: 12 января 1998 г.

Тактовая частота: 266 МГц

Количество транзисторов: 4,5 млн (0,25-мкм технология)

Ширина полосы пропускания шины: 64 бит

(Примечание: это полностью 32-разрядный микропроцессор)

Адресуемая память: ~68 Гигабайт

Тип корпуса: Tare Carrier Package (ТСР), 320 выводов

Применение: мобильные ПК и ноутбуки

Процессор Pentium (200, 233 МГц) с технологией MMX \geq

Объявлены: 8 сентября 1997 г.

Тактовая частота: 200, 233 МГц

Количество транзисторов: 4,5 млн (0,25-мкм технология)

Ширина полосы пропускания шины: 64 бит
(Примечание: это полностью 32-разрядный микропроцессор)
Адресуемая память: ~68 Гигабайт
Тип корпуса: Tape Carrier Package (TCP), 320 выводов
Применение: мобильные ПК и ноутбуки

Настольные ПК

Процессор Pentium (233 МГц) с технологией MMX \geq

Объявлен: 2 июня 1997 г.

Тактовая частота: 233 МГц (7.12 SPECint95, 5.21 SPECfp95. Показатель тестирования iCOMP \geq Index 2.0 - 203).

Количество транзисторов: 4,5 млн (0,35-мкм технология КМОП)

Ширина полосы пропускания шины: 64 бит (внешняя шина данных), 32 бит (адресная шина)

(Примечание: это полностью 32-разрядный микропроцессор)

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Число контактов: 296 (пластиковый корпус со штырьковыми выводами - PPGA)

Габариты модуля: 5 см x 5 см

Применение: высокопроизводительные настольные ПК и серверы

Процессор Pentium (200, 166 МГц) с технологией MMX \geq

Объявлен: 8 января 1997 г.

Тактовая частота: 200, 166 МГц (6.44 SPECint95, 4.87 SPECfp95.

Показатель тестирования iCOMP \geq Index 2.0 - 182).

Количество транзисторов: 4,5 млн (0,35-мкм технология КМОП)

Ширина полосы пропускания шины: 64 бит (внешняя шина данных), 32 бит (адресная шина)

(Примечание: это полностью 32-разрядный микропроцессор)

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Число контактов: 296 (пластиковый корпус со штырьковыми выводами - PPGA)

Габариты модуля: 5 см x 5 см

Применение: высокопроизводительные настольные ПК и серверы

Процессор Pentium (200 МГц)

Объявлен: 10 июня 1996 г.

Тактовая частота: 200 МГц (5.17 SPECint95, 4.32 SPECfp95. Показатель тестирования iCOMP \geq Index 2.0 - 142).

Количество транзисторов: 3.3 млн (0,35-мкм технология БиКМОП)

Ширина полосы пропускания шины: 64 бит (внешняя шина данных), 32 бит (адресная шина)

(Примечание: это полностью 32-разрядный микропроцессор)

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Число контактов: 296 (пластиковый корпус со штырьковыми выводами - PPGA)

Габариты модуля: 5 см x 5 см

Применение: высокопроизводительные настольные ПК и серверы

Процессор Pentium (166, 150 МГц)

Объявлен: 4 января 1996 г.

Тактовая частота: 166, 150 МГц (4.58 SPECint95, 3.92 SPECfp95 на системе Xpress с 1 Мб кэш-памяти второго уровня)

Количество транзисторов: 3,3 млн (0,35-мкм технология БиКМОП)

Ширина полосы пропускания шины: 64 бит (внешняя шина данных), 32 бит (адресная шина)

(Примечание: это полностью 32-разрядный микропроцессор)

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Число контактов: 296 (корпус со штырьковыми выводами - PGA)

Габариты модуля: 5 см x 5 см

Применение: высокопроизводительные настольные ПК и серверы

Процессор Pentium (133 МГц)

Объявлен: июнь 1995 г.

Тактовая частота: 133 МГц (218,9 млн. операций в секунду, 4.01 SPECint95, 3.50 SPECfp95 на системе Xpress с 1 Мб кэш-памяти второго уровня)

Количество транзисторов: 3,3 млн (0,35-мкм технология БиКМОП)

Ширина полосы пропускания шины: 64 бит (внешняя шина данных), 32 бит (адресная шина)

(Примечание: это полностью 32-разрядный микропроцессор)

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Число контактов: 296 (корпус со штырьковыми выводами - PGA)

Габариты модуля: 5 см x 5 см

Применение: высокопроизводительные настольные ПК и серверы

Процессор Pentium (120 МГц)

Объявлен: 27 марта 1995 г.

Тактовая частота: 120 МГц (203 млн. операций в секунду, 3.72

SPECint95, 2.81 SPECfp95 на системе Xpress с 1 Мб кэш-памяти второго уровня)

Количество транзисторов: 3,2 млн (0,6 и 0,35-мкм технологии БиКМОП)

Ширина полосы пропускания шины: 64 бит (внешняя шина данных), 32 бит (адресная шина)

(Примечание: это полностью 32-разрядный микропроцессор)

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Число контактов: 296 (корпус со штырьковыми выводами - PGA)

Габариты модуля: 5 см x 5 см

Применение: настольные ПК и ноутбуки

Процессоры Pentium (90 и 100 МГц)

Объявлен: 7 марта 1994 г.

Тактовая частота: 90 МГц (149,8 млн. операций в секунду, 2.74

SPECint95, 2.39 SPECfp95 на системе Gateway P5 с 256 Кб кэш-памяти второго уровня)

100 МГц (166,3 млн. операций в секунду, 3.30 SPECint95, 2.59

SPECfp95 на системе Xpress с 1 Мб кэш-памяти второго уровня)

Количество транзисторов: 3,2 млн (0,6-мкм технология БиКМОП)

Ширина полосы пропускания шины: 64 бит (внешняя шина данных), 32 бит (адресная шина)

(Примечание: это полностью 32-разрядные микропроцессоры)

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Число контактов: 296 (корпус со штырьковыми выводами - PGA)

Габариты модуля: 5 см x 5 см

Применение: настольные ПК

Процессор Pentium (75 МГц)

Объявлен: 10 октября 1994 г.

Тактовая частота: 75 МГц (126,5 млн. операций в секунду, 2.31 SPECint95, 2.02 SPECfp95 на системе Gateway P5 с 256 Кб кэш-памяти второго уровня)

Количество транзисторов: 3,2 млн (0,6-мкм технология БиКМОП)

Ширина полосы пропускания шины: 64 бит (внешняя шина данных), 32 бит (адресная шина)

(Примечание: это полностью 32-разрядный микропроцессор)

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Число контактов: 320 - корпус на ленточном носителе (TCP)

296 - плоский корпус с матрицей штырьковых выводов, расположенных в шахматном порядке (SPGA)

Габариты модуля: 5 см x 5 см

TCP: 2,4 x 2,4 см

Применение: настольные ПК и ноутбуки

Процессоры Pentium (60 и 66 МГц)

Объявлен: 22 марта 1993 г.

Тактовая частота: 60 МГц (100 млн. операций в секунду, 70.4

SPECint92, 55.1 SPECfp92 на системе Xpress с 256 Кб кэш-памяти второго уровня)

66 МГц (112 млн. операций в секунду, 77.9 SPECint92, 63.6

SPECfp92 на системе Xpress с 256 Кб кэш-памяти второго уровня)

Количество транзисторов: 3,1 млн (0,8-мкм технология БиКМОП)

Ширина полосы пропускания шины: 64 бит (внешняя шина данных), 32 бит (адресная шина)

(Примечание: это полностью 32-разрядные микропроцессоры).

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Число контактов: 273 (корпус со штырьковыми выводами - PGA)

Габариты модуля: 5,49 см x 5,49 см

Применение: настольные ПК

Процессоры Intel 486 и более ранние

Intel 486 SL CPU

Объявлен: 9 ноября 1992 г.

Тактовая частота: 20 МГц (15,4 млн. операций в секунду)

25 МГц (19 млн. операций в секунду)

33 МГц (25 млн. операций в секунду)

Адресуемая память: 64 Мбайт

Адресуемая виртуальная память: 64 терабайт

Формат процессора в мкм: 0,8

Количество транзисторов: 1,4 млн (0,8-мкм технология)

Путь доступа к внутренним данным: 32-битный

Путь доступа к внешним данным: 32-битный

Применение: первый процессор, специально предназначенный для ноутбуков

Процессор IntelDX2

Объявлен: 3 марта 1992 г.

Тактовая частота: 50 МГц (41 млн. операций в секунду, 29.9

SPECint92, 14.2 SPECfp92 на системе Micronics M4P с 256 Кб кэш-памяти второго уровня)

Процессор с тактовой частотой 66 МГц, о начале выпуска которого объявлено 10 августа 1992 г. (54 млн. операций в секунду, 39.6 SPECint92, 18.8 SPECfp92 на системе Micronics M4P с 256 Кб кэш-памяти второго уровня)

Количество транзисторов: 1,2 млн (0,8-мкм технология)

Ширина полосы пропускания шины: 32 бит

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Применение: недорогие настольные ПК

Благодаря применению технологии "удвоенной частоты" ядро микропроцессора работает в два раза быстрее шины.

Предназначен для самых дешевых настольных ПК с минимально необходимым уровнем производительности.

Intel486 SX CPU

Объявлен: 22 апреля 1991 г.

Тактовая частота: 16 МГц, начало выпуска процессора объявлено 16 сентября 1991 г. (13 млн. операций в секунду)

20 МГц (16,5 млн. операций в секунду)

25 МГц, начало выпуска процессора объявлено 16 сентября 1991 г. (20 млн. операций в секунду, 12 SPECint92)

33 МГц, начало выпуска процессора объявлено 21 сентября 1992 г. (27 млн. операций в секунду, 15.86 SPECint92)

Количество транзисторов: 1 185 000 (1 мкм); 900 000 (0,8-мкм технология)

Ширина полосы пропускания шины: 32 бит

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Применение: недорогие настольные ПК начального уровня класса Intel486 \geq .

Аналог процессора Intel486 \geq DX, но без встроенного математического сопроцессора.

Имеется возможность модернизации путем установки процессора Intel OverDrive.

Этот процессор стал типовым при работе со встроенными приложениями.

Intel386 SL CPU

Объявлен: 15 октября 1990 г.

Тактовая частота: 20 МГц (4,21 млн. операций в секунду)

25 МГц - начало выпуска объявлено 30 сентября 1991 г. (5,3 млн. операций в секунду)

Количество транзисторов: 855 000 (1 мкм)

Внутренняя архитектура: 32 бит

Ширина полосы пропускания внешней шины: 16 бит

Адресуемая память: 4 гигабайта

Виртуальная память: 64 терабайт

Применение: первый микропроцессор, специально предназначенный для портативных ПК

Высокоинтегрированная архитектура включает в себя кэш-память, шину и средства управления памятью

Intel486 DX CPU

50-кратная производительность по сравнению с процессором 8088.

Объявлен: 10 апреля 1989 г.

Тактовая частота: 25 МГц (20 млн. операций в секунду, 16.8 SPECint92, 7.40 SPECfp92)

33 МГц - начало выпуска объявлено 7 мая 1990 г. (27 млн. операций в секунду, 22.4 SPECint92 на системе Micronics M4P с 128 Кб кэш-памяти второго уровня)

50 МГц - начало выпуска объявлено 24 июня 1991 г. (41 млн. операций в секунду, 33.4 SPECint92, 14.5 SPECfp92 на системе Compaq/50L с 256 Кб кэш-памяти второго уровня)

Количество транзисторов: 1 200 000 (1 мкм, в процессорах с тактовой частотой 50 МГц применялась 0,8-мкм технология)

Ширина полосы пропускания шины: 32 бит

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Применение: настольные ПК и серверы

По сравнению с процессором 8088, производительность возросла в 50 раз

Просмотр энциклопедии Britannica с применением этого процессора занял 3,5 секунды

Intel386 SX CPU

Объявлен: 16 июня 1988 г.

Тактовая частота: 16 МГц (2,5 млн. операций в секунду)

20 МГц - начало выпуска объявлено 25 января 1989 г. (2,5 млн. операций в секунду)

25 МГц (2,7 млн. операций в секунду)

33 МГц - начало выпуска объявлено 26 октября 1992 г. (2,9 млн. операций в секунду)

Количество транзисторов: 275 000 (1,5 мкм, в дальнейшем - 1 мкм)

Внутренняя архитектура: 32 бит

Ширина полосы пропускания внешней шины: 16 бит

Адресуемая память: 4 гигабайта

Виртуальная память: 64 терабайт

Применение: настольные и портативные ПК начального уровня

Intel386 DX CPU

Объявлен: 17 октября 1985 г.

Тактовая частота: 16 МГц (от 5 до 6 млн. операций в секунду)

20 МГц - начало выпуска объявлено 16 февраля 1987 г. (от 6 до 7 млн. операций в секунду)

25 МГц - начало выпуска объявлено 4 апреля 1988 г. (8,5 млн. операций в секунду)

33 МГц - начало выпуска объявлено 10 апреля 1989 г. (11,4 млн. операций в секунду, 9.4 SPECint92 на системе Compaq/i с 16 Кб кэш-памяти второго уровня)

Количество транзисторов: 275 000 (1,5 мкм, в дальнейшем - 1 мкм)

Ширина полосы пропускания шины: 32 бит

Адресуемая память: 4 гигабайт

Виртуальная память: 64 терабайт

Применение: настольные ПК

Способен обращаться к такому объему памяти, которого достаточно для составления восьмистраничной биографической справки на каждого обитателя Земли. Просмотр энциклопедии Britannica с применением этого процессора занял 12,5 секунд

80286

Объявлен: февраль 1982 г.

Тактовая частота: 6 МГц (0,9 млн операций в секунду)

10 МГц (1,5 млн операций в секунду)

12 МГц (2,66 млн операций в секунду)

Количество транзисторов: 134 000 (1,5 мкм)

Ширина полосы пропускания шины: 16 бит

Адресуемая память: 16 мегабайт

Виртуальная память: 1 гигабайт

Применение: в то время - стандартный микропроцессор для всех моделей ПК

По сравнению с процессором 8086, производительность возросла в три-шесть раз

Просмотр энциклопедии Britannica с применением этого процессора занял 45 секунд

80186

Объявлен: 1982 г.

Примечание: применялся, главным образом, в работе с управляющими приложениями

8088

Объявлен: июнь 1979 г.

Тактовая частота: 5 МГц (0,33 млн операций в секунду)

8 МГц (0,75 млн операций в секунду)

Внутренняя архитектура: 16 бит

Ширина полосы пропускания внешней шины: 8 бит

Количество транзисторов: 29 000 (3 мкм)

Применение: стандартный микропроцессор для всех ПК производства корпорации IBM и их клонов

Аналог процессора 8086, за исключением 8-битной внешней шины

8086

Объявлен: 8 июня 1978 г.

Тактовая частота: 5 МГц (0,33 млн операций в секунду)

8 МГц (0,66 млн операций в секунду)

10 МГц (0,75 млн операций в секунду)

Количество транзисторов: 29 000 (3 мкм)

Ширина полосы пропускания шины: 16 бит

Адресуемая память: 1 мегабайт

Применение: портативные ПК

По сравнению с процессором 8080, производительность возросла десятикратно

8085

Объявлен: март 1976 г.

Тактовая частота: 5 МГц

0,37 млн операций в секунду

Количество транзисторов: 6 500 (3 мкм)

Ширина полосы пропускания шины: 8 бит

Применение: весы Toledo. Электронное взвешивание и вычисление цены товара.

Высокий уровень интеграции, впервые применен единый 5-вольтовый источник питания (ранее - 12 вольт)

8080

Объявлен: апрель 1974 г.

Тактовая частота: 2 МГц

0,64 млн операций в секунду

Количество транзисторов: 6 000 (6 мкм)

Ширина полосы пропускания шины: 8 бит

Адресуемая память: 64 Кбайт

Применение: устройства управления уличным освещением, компьютеры Altair computer (первые ПК)

По сравнению с процессором 8008, производительность возросла в десять раз, а количество микросхем поддержки уменьшилось с 20-ти до шести

8008

Объявлен: апрель 1972 г. (разрабатывался одновременно с процессором 4004)

Тактовая частота: 200 килогерц

0,06 млн операций в секунду

Количество транзисторов: 3 500 (10 мкм)

Ширина полосы пропускания шины: 8 бит

Адресуемая память: 16 Кбайт

Применение: терминалы ввода-вывода, калькуляторы общего назначения, автоматы бутылочного разлива

Обработка данных и текста

4004

Объявлен: 15 ноября 1971 г.

Тактовая частота: 108 килогерц

0,06 млн операций в секунду

Количество транзисторов: 2 300 (10 мкм)

Ширина полосы пропускания шины: 4 бит

Адресуемая память: 640 байт

Применение: калькуляторы Busicom

Первая компьютерная микросхема, арифметические вычисления

Приложение 4. Первая десятка самых мощных микропроцессорных систем (суперкомпьютеров)

№№	Производитель	Система	Год	Число процессоров
1.	IBM	ASCI White	2000	8192
2.	Compag	AlphaServer SC	2001	3024
3.	IBM	SP Power3	2001	3328
4.	Intel	ASCI Red	1999	9632
5.	IBM	ASCI Blue Pacific	1999	5808
6.	Compag	AlphaServer SC	2001	1536
7.	Hitachi	SR8000/MPP	2001	1152
8.	SGI	ASCI Blue Mountain	1998	6144
9.	IBM	SP Power3	2000	1336
10.	IBM	SP Power3	2001	1280



Микропроцессорная система ASCI White

Самый мощный суперкомпьютер современности - ASCI White, занимающий площадь размером в две баскетбольные площадки и установленный в Ливерморской национальной лаборатории. Он включает 512 SMP-узлов на базе 64-разрядных процессоров POWER3-II (в общей сложности 8192 процессора) и использует новую коммуникационную технологию Colony с пропускной

способностью около 500 Мбайт/с, что почти в четыре раза быстрее коммутатора SP high-performance switch.

Архитектура суперкомпьютера основана на зарекомендовавшей себя массивно-параллельной архитектуре RS/6000 и обеспечивает производительность в 12,3 Тфлопс (триллионов операций в секунду). Система включает в общей сложности 8 Тбайт оперативной памяти, распределенной по 16-процессорным SMP-узлам, и 160 Тбайт дисковой памяти. Доставка системы из лабораторий IBM в штате Нью-Йорк в Ливермор (Калифорния) потребовалось 28 грузовиков-трейлеров.

Все узлы системы работают под управлением ОС AIX. Суперкомпьютер используется учеными Министерства энергетики США для расчета сложных трехмерных моделей с целью поддержания ядерного оружия в безопасном состоянии. ASCI White состоит из трех отдельных систем, среди которых самой большой является White (512 узлов, 8192 процессора), а есть еще Ice (28 узлов, 448 процессоров) и Frost (68 узлов, 1088 процессоров).

Предшественником ASCI White был суперкомпьютер Blue Pacific (другое название ASCI Blue), включающий 1464 четырехпроцессорных узла на базе кристаллов PowerPC 604e/332 МГц. Узлы связаны в единую систему с помощью кабелей общей длиной почти в пять миль, а площадь машинного зала составляет 8 тыс. квадратных футов. Система ASCI Blue состоит в общей сложности из 5856 процессоров и обеспечивает пиковую производительность в 3,88 Тфлопс. Суммарный объем оперативной памяти составляет 2,6 Тбайт.

Американский национальный центр по исследованию атмосферы (NCAR) выбрал IBM в качестве поставщика самого мощного в мире суперкомпьютера, предназначенного для прогнозирования климатических изменений. Система, известная под именем Blue Sky (Синее небо) на порядок увеличивает возможности NCAR в области моделирования климата. Ядром Blue Sky являются суперкомпьютер IBM SP и системы IBM eServer p690, применение которых позволяет добиться пиковой производительности почти в 7 Тфлопс при объеме дисковой подсистемы IBM SSA в 31,5 Тбайт.



Кабельное соединение узлов суперкомпьютера ASCI White

Суперкомпьютер, получивший название "Синий шторм" (Blue Storm), создается по заказу Европейского центра среднесрочных прогнозов погоды (European Centre for Medium-Range Weather Forecasts - ECMWF). Blue Storm будет в два раза мощнее ASCI White. Для его создания необходимо 100 серверов IBM eServer p690, также известных как Regatta. Каждый системный блок размером с холодильник содержит более тысячи процессоров. В 2004 г. "Синий шторм" оснащен серверами нового поколения p960, которые делают его еще в два раза мощнее. Суперкомпьютер работает под управлением ОС AIX. Общая емкость накопителей Blue Storm составляет 1,5 петабайт, вычислительная мощность - около 23 Тфлопс. Система весит 130 т, а по мощности в 1700 раз превосходит шахматный суперкомпьютер Deep Blue.

Исследователи IBM совместно с Ливерморской национальной лабораторией ведут работы над компьютерами Blue Gene/L и Blue Gene/C. Эти компьютеры - часть начатого еще в 1999 г. с целью изучения белков 5-летнего проекта Blue Gene, в который было вложено 100 млн долл. Проектная производительность Blue Gene/L будет, таким образом, превышать суммарную производительность 500 самых мощных компьютеров в мире. При этом новый суперкомпьютер занимает площадь, равную всего половине теннисного корта. Инженеры IBM поработали и над снижением потребления энергии - его удалось уменьшить в 15 раз.