Оглавление

|  |  |  |
| --- | --- | --- |
|  |  | Стр. |
|  | Сокращения | 5 |
|  | Введение | 6 |
| 1. | Области применения микропроцессоров | 7 |
| 2. | Виды архитектур микропроцессоров | 11 |

1. 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 |

1. 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- | | 74 |
|  | разрядных МП |  |
| 5.1.5. Особенности интеграции элементов RISC- | | 78 |
|  | архитектуры в процессорах серии x86 |  |
| 5.2. МП с традиционной RISC - архитектурой | |  |
| 5.2.1. | Intel Pentium 4 | 82 |
| 5.2.2. | AMD Athlon | 96 |
| 5.2.3. | МС88110 компании Motorola | 99 |

1. Микропроцессоры с масштабируемой архитектурой

|  |  |  |
| --- | --- | --- |
| 5.3.1. | SuperSPARC | 103 |
| 5.3.2. | MicroSPARC-II | 106 |

1. 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 |

1. Микропроцессоры нетрадиционных архитектур

4

|  |  |  |
| --- | --- | --- |
| 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 |

1. Архитектуры микропроцессорных систем
2. Структуры с централизованным, децентрализованным 174 и комбинированным управлением

|  |  |  |  |
| --- | --- | --- | --- |
| 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. | МВС – 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 |

1. Многопроцессорные системы

|  |  |  |
| --- | --- | --- |
|  | 9.1. Общие требования, предъявляемые к МПС | 263 |
|  | 9.2. Классификация систем параллельной обработки дан- | 268 |
|  | ных |  |
|  | 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 | 350 |
|  | технологии |  |

5

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

микропроцессорных систем (суперкомпьютеров)

**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 элемнт ОЗУ – оперативное запоминающее устройство ЗУ – запоминающее устройство ЭВМ – электронная вычислительная машина РОН – регитр 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 комплектк БИС ЦПЭ – центральный процессорный элемент КВВ – контроллер ввода - вывода КПр – контроллер прерываний ИНС – искусственная нейронная сеть ССП – слово состояния процессора

6

**Введение**

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

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

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



7

**1. Области применения микропроцессоров**

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

* простое наращивание разрядности и производительности,
* возможность широкого распараллеливания вычислитель-ного процесса,
* эффективную обработку алгоритмов решения различных

задач,

* простоту технической и математической эксплуатации. Сама МПС, будучи оснащенной разнообразными устройст-

вами ввода - вывода (УВВ) информации, может применяться в ка-честве законченного изделия. Однако часто к МПС необходимо подавать сигналы от множества измерительных датчиков и испол-нительных механизмов какого - либо сложного объекта управле-ния 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 уст-ройствах бытового назначения. Бурно развивается производство электронных игр с использованием микропроцессоров . Они поро-ждают не только интересные средства развлечения, но и дают воз-можность 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. Наиболь-шее 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

8

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

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

(ПО).

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

Важной функцией МП является предварительная обработ-ка информации с внешних устройств (ВУ), преобразования форма-тов данных, контроллеров электромеханических внешних уст-ройств. 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 свя-зи.

Использование МП в контрольно-измерительных приборах и в качестве контрольных средств радиоэлектронных систем дает возможность проводить калибровку, испытание и поверку прибо-ров, коррекцию и температурную компенсацию, контроль и управление измерительными комплексами, преобразование и об-работку, индикацию и представление данных, диагностику и лока-лизацию неисправностей.

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

9

ра и обработки информации, где общие функции сводятся к пере-даче множества сигналов в один центр для оценки и принятия ре-шения . 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 вы-числительной системы на основе бортовой МПС.

Обобщая рассмотренные примеры использования МП, мож-но 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 парал-лельных 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 управ-ления или быть центром управляющих систем нижних контуров управления.

Использование МПС даже в простейшей схеме управления принципиально изменяет качество функционирования обслужи-ваемых им устройств. Она позволяет оптимизировать режимы ра-боты управляемых объектов или процессов и за счет этого полу-чать прямой и/или косвенный технико-экономический эффект.

Прямой технико-экономический эффект выражается в эко-номии потребляемой энергии, повышении срока службы и сниже-нии расхода материалов и оборудования. Косвенный технико-экономический эффект связан со снижением требований к обслу-живающему персоналу и повышением производительности.

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

Использование МПС повышает качество работы и произво-

10

дительность оборудования, существенно снижает требования к персоналу, работающему на нем. Цифровое управление отдельны-ми единицами оборудования на различных уровнях позволяет лег-ко собирать информацию (или вызвать ее) с нижних на верхние уровни иерархической системы управления.

Локальные системы накопления и обработки информации. Уровень управления современным предприятием или учреждени-ем требует наличия для любого специалиста или руководителя достаточно большого объема специфичной информации. Это мо-жет быть обеспечено за счет применения локальных микропро-цессорных вычислительных систем.

Локальные, т. е. расположенные на рабочем месте, МПС накопления и обработки информации экономически и технически просто осуществляют информационное обеспечение потребите-лей. Объединение локальных систем между собой в сеть и дистан-ционное подключение этой сети к центральной ЭВМ с громадным информационным архивом позволяют создать завершенную авто-матизированную систему информационного обеспечения.

Внешние устройства локальных МПС могут встраиваться в корпус ЭВМ. Их устройства образуют комплект, минимально не-обходимый для проведения вычислительных работ и обработки данных. В комплект сложных локальных МПС, ориентированных на решение инженерных и научных задач, могут входить разнооб-разные внешние устройства, например, печати, визуального ото-бражения, внешней памяти, комплексирования, пульты операторов общего назначения и т. д.

Распределенные системы управления сложными объектами. Альтернативой широко распространенным системам с централь-ным процессором становятся распределенные микропроцессорные управляющие системы. В этом случае микропроцессоры и связан-ные с ними схемы обработки данных физически располагаются вблизи мест возникновения информации, образуя локальные МПС. Такое построение системы позволяет вести обработку информа-ции на месте ее возникновения, например , вблизи двигателей, ру-лей управления, тормозной системы и т. д. В этом случае связь системы с центральной системой обработки и накопления данных и создает пространственно - распределенную систему управления.

11

В распределенных системах достигается значительный рост быстродействия получения и обработки входной информа-ции, экономия в количестве и распределении линий связи, повы-шается живучесть , существенно развиваются возможности опти-мизации режимов управления и функционирования.

Распределенные высокопроизводительные системы парал-лельных вычислений. МПС открыли новые возможности реше-ния сложных вычислительных задач , алгоритмы вычисления кото-рых допускают распараллеливание, т. е. одновременные (парал-лельные) вычисления на многих микропроцессорах.

Системы параллельных вычислений на основе десятков, со-тен и даже тысяч одинаковых или специализированных на опре-деленные задачи микропроцессоров при значительно меньших за-тратах дают такую же производительность, как и вычислительных системах на основе мощных процессоров конвейерного типа. Соз-дание МПС с большим количеством специализированных по функциональному назначению процессоров позволяет проектиро-вать 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 структу-ру памяти, системной шины, организацию ввода/вывода и т.п. Применительно к вычислительным системам термин «архитекту-ра» может быть определен как распределение функций, реали-зуемых 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,

12

управления заданиями) и системных программ (служебных, ре-дактирования, сортировки, сохранения и восстановления инфор-мации).

Архитектура второго уровня может разграничивать опре-деленные уровни внутри программного обеспечения. Например, уровень управления логическими ресурсами может включать реа-лизацию таких функций, как управление базой данных, файлами, виртуальной памятью, сетевой телеобработкой. К уровню управ-ления физическими ресурсами относятся функции управления внешней и оперативной памятью, управления процессами, выпол-няющимися в системе.

Следующий, третий, уровень отражает основную линию разграничения системы, а именно границу между системным про-граммным 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 архитекту-рой физического ввода/вывода.

При создании МП используются три наиболее широко при-меняемых вида архитектур, созданных за время их развития: реги-стровая, стековая и ориентированная на оперативную память.

**Регистровая архитектура** (архитектура типа«регистр-ре-гистр ») микропроцессора определяет наличие достаточно боль-шого набора регистров внутри больших интегральных схем (БИС) микропроцессора . Этот набор регистров образует поле сверхбыст-рой оперативной памяти (СОЗУ) с произвольной записью и вы-боркой информации.

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

13

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

Микропроцессоры с регистровой архитектурой имеют вы-сокую эффективность решения научно - технических задач, по-скольку высокая скорость работы СОЗУ позволяет эффективно использовать скоростные возможности арифметик - логического блока. Однако при переходе к решению задач управления эффек-тивность таких микропроцессоров падает , так как при переключе-ниях программ необходимо разгружать и загружать регистры СО-ЗУ.

**Стековая архитектура** микропроцессора дает возможностьсоздать поле памяти с упорядоченной последовательностью запи-си и выборки информации. Эта архитектура эффективна для орга-низации работы с подпрограммами, когда возникает постоянная необходимость перехода от текущей программы к подпрограмме, обслуживающей какое - либо ВУ, и возврат в текущую програм-му. Хранение адресов возврата позволяет организовать в стеке эф-фективную обработку последовательностей вложенных подпро-грамм.

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 систе-мы. Построение же стека большой емкости требует значительных ресурсов кристалла. Поэтому наилучшими характеристиками об-ладают МП, в которых стек реализуется вне микропроцессора - в оперативной памяти (оперативном запоминающем устройстве – ОЗУ).

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

14

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

При оценке быстродействия МП типа «память - память» не-обходимо учитывать физическую реализацию как элементов, так и связей между ними. Высокая скорость срабатывания логических элементов интегральных схем не всегда может обеспечить высо-кую скорость работы МП, поскольку большие значения индуктив-но - емкостных параметров связей на печатных платах не позво-ляют передавать сигналы без искажения. Высокий уровень техно-логии 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 высоко-скоростную совместную работу. При создании однокристальных МП регистровые СОЗУ и ОЗУ МПС имеют практически одни и те же параметры. Повышение скорости работы ОЗУ позволяет уда-лить набор регистров и стек из кристалла микропроцессора и ис-пользовать освободившиеся ресурсы для развития системы ко-манд, средств прерывания, многоразрядной обработки. Организа-ция рабочих регистров и стека в ОЗУ ведет к уменьшению скоро-сти передачи информации, однако при этом повышается общая эффективность такого решения за счет большой информационной емкости полей регистровой и стековой памяти, а также возможно-сти развития системы команд и прерываний.

Архитектура микропроцессора, ориентированная на опера-тивную память, обеспечивает экономию площади кристалла МП. В этом случае на кристалле размещается только регистр - указа-тель начального файла набора регистров. Адресация остальных ре-гистров осуществляется указанием в команде специальным указа-телем - кодом смещения. Доступ к рабочим регистрам в этом слу-чае замедляется, поскольку приходится совершать сопряженное с затратами времени кольцевое «путешествие» из процессора во внекристальную память, где размещаются рабочие регистры. Од-нако контекстное переключение в микропроцессоре с такой архи-тектурой происходит быстро, поскольку при прерывании необхо-

15

димо только изменить значение содержимого регистра - указате-ля рабочей области памяти.

Другая отличительная особенность архитектуры МП, ори-ентированной на оперативную память - двухадресный формат ко-манд. В этих МП нет специального накапливающего регистра , вы-полняющего функции подразумеваемой ячейки результата для всех двухоперандных команд. Результат формируется в соответст-вии с алгоритмом, приведенном для примера на рис. 1,а, где опе-рация 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 коман-де. В то же время в процессорах с одноадресной регистровой ар-хитектурой для достижения той же самой цели приходится ис-пользовать две команды:

* + команду пересылки операнда Y во внутренний регистр Рг,
  + команду сложения содержимого внутреннего регистра Рг
* содержимым ячейки памяти X и пересылки результата в ячейку

X (рис. 1,б).

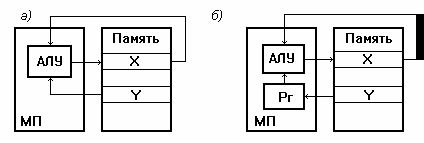


Рис. 1

В первом случае при компиляции программ для компилято-ров 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 архитекту-рой, ориентированной на оперативную память, позволяет им легко

16

находить применение в МПС, работающим в реальном масшта-бе времени.

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

Распределенные системы управления часто требуют приме-нения полуавтономных контроллеров , которые должны вписы-ваться в определенные иерархические структуры. При этом архи-тектура МП, ориентированная на память, обеспечивает естествен-ный и эффективный интерфейс между контроллерами, располо-женными на одном иерархическом уровне, и процессорами управ-ления, 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). Раз-работкой и производством МП в то время занимались ряд фирм США, среди которых наиболее совершенной технологией на фоне других обладала фирма Intel. Поэтому историю развития рассмот-рим на примере развития МП данной фирмы.

До начала 70-х годов вычислительные машины были дос-тупны весьма ограниченному кругу специалистов, а их примене-ние, как правило, оставалось окутанным завесой секретности и ма-ло известным широкой публике. Однако в 1971 г. произошло со-бытие, которое резко изменило ситуацию и превратило вычисли-тельную машину в повседневный рабочий инструмент десятков миллионов людей.

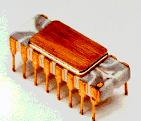
17

В 1971 году фирма Intel из небольшого американского го-родка Санта-Клара (шт. Калифорния) создала новый полупровод-никовый 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 предпри-ятия было создание полупроводниковых запоминающих уст-ройств, призванных заменить ЗУ на магнитных сердечниках. По-скольку к концу 60-х годов память этого типа практически исчер-пала весь свой потенциал развития, проблема была весьма акту-альной, 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 япон-ской фирмы Busicom. В результате его реализации был разработан кристалл в сопровождении соответствующих средств поддержки.

**МП i4004.** 15ноября1971г. Intelприступила к поставкампервого в мире микропроцессора Intel 4004 - именно такое обозна-чение получил первый прибор, послуживший отправной точкой абсолютно новому классу полупроводниковых устройств.

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



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

18

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 реги-стровый файл центрального процессора (ЦП) насчитывал шестна-дцать 4-разрядных регистров, которые можно было использовать и как восемь 8-разрядных (восемь 4-разрядных пар). Такая органи-зация РОН сохранена и в последующих микропроцессорах фирмы Intel. Три регистра стека обеспечивали три уровня вложения под-программ.

МП i4004 монтировался в пластмассовый или металлокера-мический корпус типа DIP (Dual In-line Package) всего с 16 выво-дами.

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

Вместе с тем кристалл располагал весьма ограниченными средствами ввода/ вывода, а в системе команд отсутствовали опе-рации 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)

* обработки прерываний. Это объясняется тем, что в калькулято-рах, где поначалу и планировалось использовать прибор, особой необходимости в этих средствах нет.

Цикл команды процессора состоял из восьми тактов за-дающего генератора. Как уже отмечалось, МП i4004 монтировался в корпус всего с 16 выводами - самый распространенный (а значит,

* 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

19

шины адреса (ША) и данных, причем 12 - разрядный адрес вы-давать порциями по четыре разряда, что, конечно, не могло не ска-заться на длительности машинного цикла. Прием команды по та-кому интерфейсу требовал еще двух тактов. На исполнение же са-мой 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 вспомо-гательных микросхем. Поскольку все они были рассчитаны на со-вместное использование, разработка аппаратных средств системы заметно упрощалась.

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

**МП i4040**.Вi4040сохранены все функциональные возмож-ности предшествующей модели и существенно улучшены как тех-нические, так и программные средства. Система команд пополни-лась 14 инструкциями , включая выполнение логических операций И и ИЛИ; кроме того, в процессор были введены средства остано-ва и обработки прерываний.

Претерпела некоторые изменения и архитектура устройст-ва. 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 об-ращаться к двум блокам памяти команд объемом 4 Кбайт и за ка-ждым из них программист мог закрепить свою область регистров. Наряду с этим восемь РОН были всегда доступны для использова-ния. В итоге получилась достаточно гибкая и удобная структура, позволявшая разрабатывать самостоятельные программные моду-ли, способные взаимодействовать через общую часть регистрового файла.

20

Обработка одноуровневых прерываний является одним из наиболее существенных новшеств МП i4040. Эта функция превра-тила его в полноценный процессор и сделала возможным исполь-зование в системах реального масштаба времени. Благодаря при-менению сигнала «останов» стала реальностью синхронизация ра-боты 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 эффек-тивной системы команд. 60 инструкций, ориентированных на ши-рокий 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.** Intelс1апреля1972г.начала поставки первого вотрасли 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 ра-ботами над i4004. Кристалл изготавливался по p-канальной МОП-технологии с проектными нормами 10 мкм и содержал 3500 тран-зисторов. Процессор работал на частоте 500кГц при длительности машинного цикла 20 мкс (10 периодов задающего генератора).

* отличие от своих предшественников новый МП допускал применение комбинации ПЗУ и ОЗУ. Помимо увеличения разряд-ности и перехода на использование общего поля памяти для ко-манд и данных, структура процессора претерпела еще ряд сущест-венных изменений. Прежде всего это коснулось регистрового фай-ла 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 па-мяти.
* связи с этим следовало бы ожидать снижения производи-тельности, которого на самом деле не произошло, поскольку опе-рации с памятью i8008 выполнял быстрее предыдущих моделей благодаря меньшему количеству состояний в машинном цикле и отсутствию необходимости исполнения минимум трех подготови-

21

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

Вместе с тем объем блока регистров был ограничен воз-можностями 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 дли-тельность 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 медлен-ными 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 использо-вать 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 ус-ловных переходов, а также логических инструкций и команд сдви-га. Новый кристалл мог адресовать память объемом до 16 Кбайт (объем ЗУ для МП типа i4040 не превышал 8 Кбайт). Его произво-дительность по сравнению с 4-разрядными системами возросла в

2,3 раза.

Процессор с такими параметрами уже можно было приме-нять 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 использо-вать их шифрацию, что повлекло за собой необходимость установ-ки дополнительного внешнего оборудования для формирования сигналов управления. В среднем для сопряжения процессора с па-

22

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

Вскоре после выхода i8008 появилась его усовершенство-ванная 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 процес-сора наряду с большим (по тому времени) объемом оперативной памяти послужило лучшей рекомендацией для активного исполь-зования кристалла в различных областях, начиная от промышлен-ности 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 начи-нала не удовлетворять многих разработчиков. Кроме того, некото-рые области применения настойчиво требовали расширения не только количества, но и номенклатуры периферийных устройств. Системщики уже с трудом могли обходиться без такой традици-онной 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 p-канальной МОП -технологии для создания сложных высокопроизводительных МП были уже практически ис-черпаны, поэтому направление главного удара перенесли на тех-нологию n-МОП. Перед проектировщиками стояли не менее слож-ные проблемы - разработка эффективной системы команд, рассчи-танной на широкий круг решаемых задач, при сохранении про-граммной совместимости с предыдущей моделью, расширение объема адресуемой памяти, поддержка интенсивного ввода/ вывода без существенной потери производительности процессора, совер-шенствование подсистемы обработки прерываний. Указанные причины привели к созданию нового прибора – i8080.

**МП i8080** -триумф8-разрядных систем,который появился1 апреля 1974 г. Благодаря использованию технологии n-МОП с

23

проектными нормами 6 мкм, на кристалле удалось разместить 6 тыс. транзисторов. При этом геометрические размеры самого кри-сталла по сравнению с i8008 увеличились незначительно. Следова-тельно, процент выхода годных изделий и ряд экономических по-казателей производства, включая себестоимость, удалось сохра-нить 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 пред-шествующей модели. Объем памяти, адресуемой процессором, был увеличен в четыре раза и достиг 64 Кбайт (кстати, в то время ОЗУ такой емкости предлагали потребителям минимальные кон-фигурации многих мини-ЭВМ). В сочетании с эффективным ме-ханизмом обработки прерываний это давало им возможность ши-рокого применения нового МП в сложных системах сбора и обра-ботки информации различного назначения, особенно функциони-рующих в реальном масштабе времени. За счет использования корпуса с 40 выводами удалось разделить адресную и информаци-онную шины процессора, в результате отпала необходимость при-менения дополнительных внешних схем для разделения потоков адресов и данных. Общее же количество микросхем, требовавших-ся для построения системы в минимальной конфигурации, сокра-тилось с 20 до 6, т. е. более чем в три раза. В регистровый файл были введены указатель стека , активно используемый при обра-ботке 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 со-став РОН нового процессора были включены основные рабочие регистры предыдущей модели. Правда, полной совместимости с i8008 достичь не удалось, так как процедуры обращения к подпро-граммам и инструкции ввода/вывода МП i8080 в значительной степени отличались от соответствующих процедур и операций

24

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

Включение в систему команд ряда инструкций, адресую-щих память с использованием трех пар регистров (в i8008 для это-го выделялась одна пара), придало дополнительную гибкость. Реа-лизация 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 - логи-ческого устройства упростило схему управления внутренней ши-ной, поскольку при этом отпала необходимость в ее использова-нии 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 техниче-ское решение позволило довести общее число источников преры-ваний в системе до 256. Правда, до появления специализирован-ных БИС контроллеров прерываний схема формирования векторов прерываний требовала применения до десяти дополнительных чи-пов 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 универ-сальных 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 послед-ним) ударом микропроцессоров по большим системам. ПДП от-крыл 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 полноцен-ную вычислительную систему.

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

25

контроллера ПДП и контроллера прерываний. Благодаря хорошо продуманному составу комплекта, проектирование МПС на его ба-зе 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 получение16-разрядного микропроцессора с производительностью, на поря-док превышающей аналогичный параметр кристалла i8080 и по-зволяющего создавать многопроцессорные системы. Поставленная задача решалась за счет дальнейшего совершенствования архитек-турных 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 (H-MOS), позволившей разместить на кристалле 29 тыс. транзисторов. Высокое быстро-действие 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, един-ственной, но вместе с тем и исключительно важной характеристи-кой, 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 одновре-менно. В результате исполнение одной команды совмещалось во времени с выборкой следующей инструкции или данных из памя-ти. Более того, в МП появился регистровый файл команд, что да-вало дополнительную экономию времени при обращениях к памя-ти. Алгоритм работы операционного и интерфейсного блоков по-зволял вести обработку команд, находящихся в конвейере команд

26

регистрового файла, одновременно с их вводом из программной памяти.

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

Большая емкость ОЗУ позволяла перевести проекты по-строения сложных операционных и прикладных систем из области теории в сферу практической реализации . Наряду с поддержкой ввода/вывода по каналу прямого доступа к памяти i8086 обеспечи-вал адресацию до 64К портов программно-управляемого вво-да/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 одно-процессорных системах и предполагал работу кристалла без БИС контроллера шины. Максимальный режим был ориентирован на применение МП в многопроцессорных системах и требовал нали-чия указанного контроллера. Таким образом, один и тот же про-цессор с одинаковым успехом мог применяться в системах раз-личного класса.

Система команд процессора содержала 147 инструкций. Она позволяла решать задачи управления практически любой сложности. Появление среди них таких операций, как умножение

* деление 16-разрядных чисел со знаком и без знака, команд обра-ботки массивов данных, а также программно-управляемых преры-ваний дает все основания назвать этот кристалл универсальным, рассчитанным на использование не только в сложных контролле-рах, но и в качестве центрального процессора ЭВМ общего назна-чения.

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

* системного ПО.

Использование микросхем i8086 в персональных компью-терах IBM предопределило дальнейшее развитие корпорации Intel

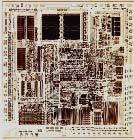
27

как разработчика и изготовителя универсальных процессоров общего назначения.

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

**3.2. Микропроцессоры второго поколения**

**МП i80286** второго поколения былразработан к 1 февраля 1982 г. Оставшись 16-разрядным прибором, по производительности новый ЦП в 3 - 6 раз превзошел своего пред-шественника при тактовой частоте первой модификации 8 МГц. Благодаря использова-нию многовыводного корпуса разработчики смогли применить схему с раздельными ши-нами адресов и данных.



24 разряда адреса позволили обращаться к физической па-мяти объемом до 16 МБайт. Встроенная система управления памя-тью и средства ее защиты открывали широкие возможности ис-пользования 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 не-сколько 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 пре-имущества МП в полном объеме, и, прежде всего, большой объем основной памяти, дающий возможность работать ему в многоза-дачном варианте. Ведь основная проблема многозадачности была

28

в том, что предыдущие модели МП исполняемые программы могли быть записаны по любому адресу памяти, даже в занятые ячейки памяти ранее исполнявшимися программами . Операцион-ная 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 контро-лировала все действия программ. Но для этого пришлось отказать-ся от компиляции приложений в готовые машинные коды - они стали интерпретируемыми , а производительность упала раз в два-дцать. 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 ре-альном режиме, но программистам пришлось использовать поня-тие «логического адреса». Логический адрес состоял из 32 бит: се-лектора (16 бит) и смещения (16 бит) (рис. 2). При этом в сегмент-ных регистрах теперь хранился не сегмент, а селектор. Селектор - это индекс в таблице дескрипторов.

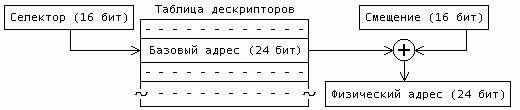


Рис. 2

Запись в таблице дескрипторов содержала всю необходи-мую информацию о некотором блоке памяти: его базовый адрес, размер всего блока, его тип (код или данные) и сведения о приори-тете программы - владельца. Таким образом, каждый дескриптор полностью описывал один сегмент программы. Размер этого деск-риптора 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.

29

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 про-грамме были доступны глобальная и локальная (своя) таблицы де-скрипторов. 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 недос-татками: несмотря на возможность адресовать 16 Мб памяти, мак-симальный 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 приложе-ний способ перемещения данных операционной системой из памя-ти на жесткий диск - для реализации этого программам приходи-лось прибегать к разным ухищрениям вроде «запирания» и «отпи-рания» указателей на блок памяти.

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

Указаные недостатки и высокая потребность в высокопро-изводительных 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, характер-ные для многозадачных сред.

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

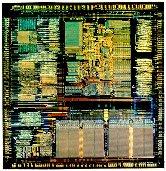
30

дачных ОС требовало усовершенствования устройства управле-ния памятью.

**3.3. Микропроцессоры третьего поколения**

**МП i80386** третьего поколения был представлен17октября1985 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 памя-ти объемом до 4 Гбайт и виртуальной памяти емкостью до 64 Тбайт. Встроен-ная в МП система управления памятью и защиты включала регистры преобразова-ния адреса, механизмы защиты опера-тивной памяти и улучшенные аппарат-ные средства поддержки многозадачных ОС.



Помимо работы с виртуальной памятью допускались операции с памя-тью, имевшей страничную организацию. Предварительная выбор-

ка команд, буфер на 16 инструкций, конвейер команд и аппаратная реализация функций преобразования адреса значительно умень-шили 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. Безуслов-но, новый прибор остался совместимым со своими предшествен-никами на уровне объектных кодов.

Одной из наиболее любопытных особенностей рассматри-ваемой разработки компании было использование высокоскорост-ной кэш-памяти, позволившей существенно повысить производи-тельность систем на базе 386-го процессора (еще один атрибут универсальных машин, который стал применяться в микропроцес-сорных системах). Для управления работой этой памятью была разработана БИС высокопроизводительного контроллера кэш-

31

памяти типа i82385, с помощью которой формировался двухвхо-довой множественный ассоциативный кэш. Указанная БИС обес-печивала 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 кристал-ла - 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 ко-манд и регистрами.

Одно из основных ограничений реального режима на прак-тике было связано с предельным объемом адресуемой памяти, равным 1 Мбайт. От него свободен защищенный режим, позво-ляющий воспользоваться всеми преимуществами архитектуры но-вого МП. Размер адресного пространства в этом случае увеличи-вался до 4 Гбайт, а объем поддерживаемых программ - до 64 Тбайт.

Производителям ПО это позволяло задействовать достаточ-но гибкие методы разработки и создавать более крупные про-граммные пакеты. Для конечных пользователей выполнение при-ложений, рассчитанных на работу в реальном и защищенном ре-жимах, происходило без каких-либо функциональных отличий, поскольку управление обоими режимами базировалось на средст-вах ОС и специальном прикладном ПО. Однако системы защи-щенного режима обладали более высоким быстродействием и воз-можностями организации истинной многозадачности.

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

32

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 при-кладным задачам использование различных областей памяти. Бла-годаря таким возможностям аппаратуры, можно было вместо не-скольких МП типа i8086 использовать один процессор i80386, со-хранив 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 ка-тегорий 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 че-рез 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 прибли-жалась к 1 млн. транзисторов на кристалле и 10 апреля 1989 г. Intel объявила о начале выпуска 32-разрядного прибора четвертого по-коления - i80486, ставшего после устройств i8080 и i8086 еще од-ним долгожителем.

**3.4. Микропроцессоры четвертого поколения**

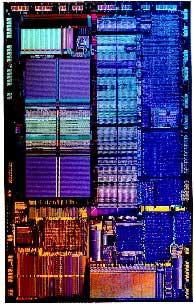
**МП i80486.** Архитектура нового МП отчасти напоминаластроение своего предшественника, но вместе с тем имела и ряд ко-ренных отличий. 1,2 млн. транзисторов позволили разработчикам реализовать на кристалле быстродействующую кэш-память (L1) и математический сопроцессор. Такое техническое решение свело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

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

33

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 тре-бовал 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 по3бита статистики использования. Алгоритм LRU (Least Recently Used) основан на поиске элемента, к которому дольше всего не было обращений. При каждом обраще-нии к строке кэш-контроллер увеличи-вал на 1 соответствующий счетчик LRU. Приставка «псевдо» означает лишь несовершенство механизма рабо-ты, ведь под счетчик отводилось всего 3 бита, что дает всего 8 состояний счет-чика (23). После 8-го обращения к стро-ке счетчик обнулится и соответствую-щая строка из самой «необходимой» станет самой «не необходимой» и будет



прямым кандидатом на замещение . Кэш-память первых 486-х рабо-

тала 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 да-вало 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 присутст-вовала 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

34

функционирует быстрее второго. Это связано с тем, что при чте-нии данных из кэш-памяти L2 процессор все равно вынужден де-лать 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 дан-ных и для команд.

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

Ровно через два года после выпуска i80486 появилась уп-рощенная версия кристалла (без сопроцессора), получившая обо-значение i80486SX. Дальнейшее совершенствование пошло по пу-ти увеличения тактовой частоты: были представлены версии на 50, 66, 75 и 100 МГц.

**3.5. Микропроцессоры пятого поколения**

**Pentium Р5.** Выпуск высокопроизводительных МПPentiumР5 началось 22-го марта 1993 года. Это был первый процессор с двухконвейерной структурой . Он имел так-товые 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 разъ-

ема Socket 4.

Как показали результаты эксплуатации МП, простое наращи-вание тактовой частоты неоднозначно влияло на увеличение его

35

производительности. В первую очередь это связано с большим разбросом во времени выполнения различных программ (прило-жений), связанных с разницей в производительности их различных компонентов. Так, например, разброс высоких и низких значений производительности относительно среднего значения компьюте-ров, построенных на процессорах 80486 и Pentium Р5, составля-ет ориентировочно от 10 % до 20 % (рис. 3).

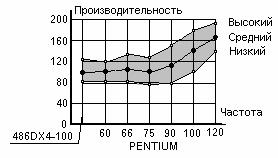


Рис. 3

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

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

В марте 1994 года Intel выпустила МП **Pentium P54**. Про-цессор 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 ин-струкции). 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

36

млн. транзисторов. Выпускался для разъема Socket 5, позднее

Socket 7.

**Pentium MMX** (**P55**)был выпущен8января1997года.Онпришел на смену МП Р54 в связи с появлением все большего числа мультимедийных приложений. В нем был реализован новый набор из 57 команд MMX (Multi Media eXtention), существенно уве-



личивающий 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 Р54 в нем был вдвое увеличен размер L1, который соста-вил 32 Кб. Как и в предыдущих версиях L1 был разбит на два бло-ка 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 разъе-

мом Socket 7.

**3.6. Микропроцессоры шестого поколения**

**Pentium PRO (Р6)** был выпущен1ноября1995года.Отпредыдущего поколения их отличало применение технологии ди-намического 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.

37

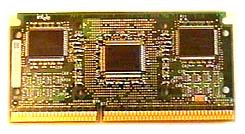
Процессоры Pentium PRO выпускались в корпусах SPGA (Staggered Pin Grid Array) с матрицей штырьковых выводов. В од-ном корпусе было установлено два кристалла – ядро процессора и L2 собственного изготовления. Устанавливался в Socket 8 с воз-можностью объединить до 4-х процессоров для симметричной мультипроцессорной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

При 32-битных вычислениях и многозадачности значитель-но 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. тран-зисторов.

**Pentium II** (**Klamath**)появились7мая1997года.Эти про-цессоры 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, реа-

лизующих L2, и вспомогательные дискретные элементы (резисто-ры и конденсаторы).

Такой подход можно считать шагом назад – у Intel уже бы-ла отработана технология встраивания в ядро кэша второго уров-ня. Но таким образом можно было использовать микросхемы па-мяти сторонних производителей. В свое время, Intel считала такой подход перспективным на ближайшие 10 лет, хотя через непро-должительное время отказалась от него.

38

В то же время сохранилась независимость шины L2, ко-торая 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 час-тоте 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 сис-темной шины 66 МГц. При этом L2 работал на половинной часто-те 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 позво-ляет объединять лишь пару процессоров для реализации симмет-ричной мультипроцессорной системы, либо системы с избыточ-ным контролем функциональности.

26 января 1998 году вышел процессор из линейки **Pentium** **II** (**Deschutes)**.ОтKlamathотличался более тонким технологиче-ским процессом – 0.25 мкм и частотой шины 100 МГц. Имел так-товые 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,вышедший11августа1998 года, был предназначен для замены Pentium PRO на старых материнских платах . Он носил кодовое имя **P6T**.Имел частоту333МГц.Кэш первогоуровня – 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 вет-

кой в направлении технологии микропроцессоров, направленной

39

на удешевление своей продукции. Он был выпущен как альтер-нативный вариант Pentium II, имевший довольно высокое соотно-шение «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 произ-водительности, 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 выпускал-ся без защитного картриджа - конструктив – SEPP (Single Edge Pin Package). Разъем - Slot 1.

Начиная с частоты 300 МГц, появились процессоры **Celeron (Mendocino)** с интегрированным в ядроL2,работающимна 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

(300A - 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 интегрированно-го высокоскоростного L1. Так как L1 был интегрирован в ядро, значительно увеличилось количество транзисторов, из которых со-стоит процессор - 19 млн. Техпроцесс остался прежним – 0.25 мкм.

Для мощных систем Intel выпустил 29 июня 1998 года МП **Pentium II Xeon** -серверный вариант процессораPentium II,при-шедший 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-

40

пускался в конструктиве SECC для Slot 2. Это тоже краевой разъем, но с 330 контактами, регулятором напряжения VRM, за-поминающим устройством EEPROM. Способен работать в муль-типроцессорных конфигурациях.

L2, как и в Pentium PRO, полноскоростной. Только здесь он находится на одной плате с процессором, а не интегрирован в яд-ро. 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. транзи-сторов.

**Pentium III (Katmai)** был разработан к26февраля1999го-да и мало чем отличался от Pentium II. Он работал на такой же ши-не 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 яв-ляется 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 выпущен25октября1999

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

41

новная масса нововведенных транзисторов относится к интегри-рованному 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

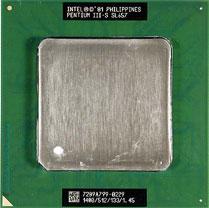


только повышала стоимость процессора. Поэтому вскоре процес-соры стали выходить в конструктиве FC-PGA (Flip-Chip PGA). Как и Celeron Mendocino, они работали в разъеме Socket 370.

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

**Pentium III (Tualatin)** пришел на сменуCoppermine 21ию-

ня 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 за-полнить 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 проек-та.

Изменений в самом ядре немного - добавилась только тех-нология "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 теплорассеивающей пла-стиной, которая также защищает его от повреждения при установ-ке радиатора.

42

МП Tualatin работали на шине с частотой 133 МГц и со-стояли из 44 млн . транзисторов. Поддерживали наборы инструк-ций 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 вари-ант).

**Celeron (Coppermine Lite)** были разработаны29марта

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 техноло-гической 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 рабо-тали 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 инструк-ции). Процессор состоял из 28.1 млн. транзисторов.

2-го октября 2001 года, Intel пе-реводит 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

43

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

**Pentium III (Tanner)** был построен на базеPentium IIIKatmai. Содержал 512, 1024 или 2048 Кб полноскоростной кэш памяти второго уровня. L1 - 16 Кб + 16 Кб. Выпускался с частота-ми 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 бо-лее) 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 сис-темной шины 100 и 133 МГц ( в зависимости от версии). Выпуска-лись 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. транзи-сторов

**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 технологиче-ских 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. По-этому компания провела доработку алгоритмов предсказания пе-

реходов (Advanced Dynamic Execution).

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

44

Pentium 4 имел всего 8 Кб кэш данных. Команды , сохраняются в так называемом Trace Cache. Там они хранятся уже в декодиро-ванном виде, т.е. в виде последовательности микроопераций, по-ступающих для выполнения в исполнительные устройства процес-сора. Емость этого кэша составляет 12000 микроопераций.

В новом процессоре был расширен набор команд - SSE2. К 70 инструкциям SSE, добавились еще 144 новые инструкции. Од-ной 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. Шири-на 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 ориентирова-на в первую очередь на рост частоты, то первые процессоры Pentium 4 показали крайне низкую производительность. В боль-шинстве задач 1.4 ГГц процессор уступал Pentium III Coppermine, работающему на частоте 1000 МГц.

27 августа 2001 года, появились МП **Willamette** предназна-ченные для установки в новый разъем - Socket 478. Процессор по-вторял все характеристики своего предка, за исключением конст-руктива - mPGA и разъема Socket 478.

Размеры процессора уменьшились благодаря тому, что те-перь выводы сделаны непосредственно под ядром процессора.

45

Этот процессор, как и предшественник, работал на частотах от

1.4 до 2.0 ГГц.

**Pentium 4 Northwood** –так называется следующее ядро,накотором 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 количе-ство транзисторов, которые составляют процессор – теперь их ста-ло 55 млн. Естественно, что осталась поддержка наборов инструк-

ций 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.

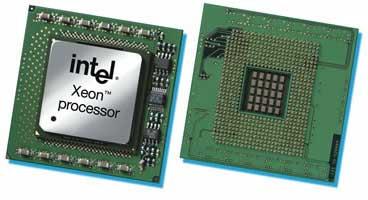
6-го мая 2002 года, Intel выпустила процессор на базе ядра Northwood с частотой системной шины 533 МГц (4×133 МГц) и тактовой частотой 2.26 ГГц. Так как модели с частотой шины 400 МГц выпускались с частотами до 2.6 ГГц, то и тут была применена буквенная маркировка.

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 часто-тами – от 2.4 ГГц.

**Pentium 4 XEON** были представленыIntel 21мая2001года,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 Xeon имеет все то, что есть в Pentium 4 – как достоинства новой архитектуры, так и ее недостатки.

Первые модели Xeon 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.

46



Кэш память первого уровня, как и у всех процессоров ли-нейки 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 но-

вом Xeon применена 400 МГц системная шина (4×100 МГц) кото-рая синхронно работает с двумя каналами памяти на частоте 400 МГц.

Исторически, линейки процессоров Intel Xeon (то есть

Pentium II Xeon, Pentium III Xeon) всегда использовали отличный от обычных версий процессора конструктив. В то время как про-цессоры Pentium II и Pentium III выпускались в 242-контактном Slot 1 варианте, то их Xeon версии использовали 330-контактный разъем Slot-2. Большинство добавочных ножек использовалось для снабжения кристалла дополнительной энергией.

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

Процессор мог работать только в одно- или двухпроцессор-ных конфигурациях.

9 января 2002 года появляются процессоры Xeon, сделан-ные на базе ядра Northwood с применением 0.13 мкм. техпроцесса, и оснащенные 512 Кб кэш памяти второго уровня. Кодовое назва-

ние ядра – **Prestonia**.

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

47

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

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

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

HT – это не настоящая многопроцессорность, ведь количе-ство блоков непосредственно исполняющих команды не измени-лось. 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 оп-тимизированные приложения могут, к примеру, обращаться к дан-ным, которых нет в кэш-памяти процессора, заставляя его про-стаивать. Если сама архитектура NetBurst была рассчитана на по-вышение количества мегагерц, то Hyper-Threading наоборот, рас-считан 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 поддер-живающей новую технологию. Также технологию должен под-держивать чипсет и BIOS системной платы.

48

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, изго-товленные 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 интегриро-ванной L3 размером 1 или 2 Мб. Благодаря этому увеличилось ко-личество транзисторов, составляющих процессор до 108 млн.

18 ноября 2002 года появились процессоры Xeon работаю-щие 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 выпускаться14ноября2002года.Онимеет частоту 3.06 ГГц, а системная шина тактировалась частотой 533 МГц с поддержкой технологии Hyper-Threading.

**Celeron** (**Willamette 128)** был выпущен с целью вытесне-ния 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, по-строенный 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 систем-ную 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.

49



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 уре-занной до 128 Кб кэш - памятью второго уровня). Первым процес-сором на этом ядре был Celeron 2.0 ГГц, который вышел 18 сен-

тября 2002 года. Он, как и Celeron Willamette 128, полностью по-

вторяет 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 частот-ный потенциал, поэтому запас для разгона есть.

* конце 2003 года Intel представила новое ядро для своих процессоров – **Prescott**. Эти процессоры изготовлены с примене-нием 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.
* МП добавлен новый набор инструкций (или расширен уже присутствующий), включающий 15 новых инструкций по пе-реводу чисел с плавающей запятой в целые, арифметику ком-плексных чисел, специальные команды для декодирования видео, SIMD-инструкции для формата с плавающей запятой и процесс синхронизации потоков.

50

Первые процессоры с этим ядром предназначены для ра-боты на частотах 3.2 и 3.4 ГГц. Их корпуса совместимы с корпу-

сами процессоров Pentium 4 Northwood.

На базе нового ядра продолжен выпуск процессоров линей-ки Celeron. Чипы **Celeron** на ядре **Prescott** быстрее предшествен-ников 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 технологиейHyper-Threading, работает на системной шине 800 МГц, имеет тактовую частоту ядра 3.2 ГГц. Но главным его отличием от предшествую-щих Pentium 4 стало наличие интегрированной в кристалл кэш-памяти третьего уровня L3 объемом 2 Мб. Эта кэш-память допол-няет стандартный кэш L2 512 кбайт и работает также на частоте ядра процессора (правда, с гораздо большей латентностью, по-скольку 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 интегриро-ванной кэш-памятью L3. Его немного изменили с целью поддерж-ки системной шины 800 МГц, уменьшения энергопотребления и др. и упаковали в стандартный корпус от Pentium 4.

В настоящее время Intel ведет активные работы по созда-нию следующих поколений кристаллов с проектными технологи-ческими нормами 65 нм. Также ведутся разработки и есть рабо-тающие 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

51

системную шину 1200 МГц и позволяющее получить рабочую частоту свыше 10 ГГц. Nehalem будет основан на совершенно но-вой архитектуре. Это будет не модернизированный Pentium 4, как Prescott и Tejas. В нем будет применена система аппаратной защи-ты LaGrande, и по некоторым данным, использована более совер-шенная технология многопоточной обработки. Число транзисто-ров 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 ал-горитмом. Для решения широкого круга задач в различных облас-тях применений микропроцессор должен обладать алгоритмически полной системой команд (операций).

Теоретически показано, что минимальная алгоритмически полная система команд процессора состоит из одной или несколь-ких универсальных команд. Однако использование процессоров с минимальными по числу операций системами команд ведет к не-экономичному использованию информационных емкостей памяти и значительным затратам времени на выполнение «длинных» про-грамм. Поэтому обычно в МП встраиваются аппаратурные средст-ва, позволяющие реализовать многие десятки и сотни команд. Та-кие развитые системы команд дают возможность обеспечить ком-пактную запись алгоритмов и соответственно эффективные про-граммы.

При проектировании МП решаются задачи определения на-боров команд, выполняемых программным или аппаратурным способом на основе заданной системы микрокоманд. Аппаратур-ная реализация сложных команд дает возможность увеличить бы-стродействие микропроцессора, но требует значительных аппара-турных ресурсов кристалла интегральной схемы МП. Программ-ная реализация сложных команд позволяет обеспечивать програм-мирование сложных задач, изменять количество и особенности исполнения сложных команд. Однако скорость исполнения про-

52

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 периферий-ными микропроцессорами. Эти расширяющие возможности обра-ботки данных специальные арифметические или логические МП позволяют ускорить выполнение определенных команд и тем са-мым сократить время исполнения программ.

Для описания МП как функциональных устройств необхо-димо охарактеризовать формат обрабатываемых данных и команд, количество, тип и гибкость команд, методы адресации данных, число внутренних регистров общего назначения и регистров ре-зультата, возможности организации и адресации стека, параметры виртуальной памяти и информационную емкость прямо адресуе-мой памяти. Большое значение имеют средства построения систе-мы прерываний программ, построения эффективных систем ввода

— вывода данных и развитого интерфейса.

МП могут быть реализованы на различной физической ос-нове: на электронной, оптоэлектронной, оптической, биологиче-ской и даже на пневматической или гидравлической.

**По назначению** различают универсальные и специализиро-ванные микропроцессоры.

Универсальные МП предназначены для решения широкого круга задач. При этом их эффективная производительность слабо зависит от проблемной специфики решаемых задач В системе ко-манд МП заложена алгоритмическая универсальность, означаю-щая, что выполняемый машиной состав команд позволяет полу-чить преобразование информации в соответствии с любым задан-ным алгоритмом.

К универсальным МП относятся и секционные микропро-цессоры, поскольку для них система команд может быть оптими-зирована в каждом частном проекте создания секционного микро-процессора.

53

Эта группа МП наиболее многочисленна, в нее входят та-

кие комплекты как К580, Z80, Intel 80×86, К582, К587, К1804, К1810 и др.

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

Специализированные МП имеют ориентацию на ускорен-ное выполнение определенных функций, что позволяет резко уве-личить эффективную производительность при решении только оп-ределенных задач.

Среди специализированных микропроцессоров можно вы-делить различные микроконтроллеры, ориентированные на вы-полнение сложных последовательностей логических операций; математические МП, предназначенные для повышения производи-тельности при выполнении арифметических операций за счет, на-пример матричных методов их выполнения; МП для обработки данных в различных областях применений и т. д.

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

**По виду обрабатываемых входных сигналов** различаютцифровые 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 преобразова-тель в цифровой форме, обрабатываются и после обратного преоб-разования в аналоговую форму поступают на выход.

* архитектурной точки зрения такие микропроцессоры представляют собой аналоговые функциональные преобразователи

54

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

Обычно в составе однокристальных аналоговых МП имеет-ся несколько каналов аналого-цифрового и цифро-аналогового преобразования. В аналоговом микропроцессоре разрядность об-рабатываемых данных достигает 24 бит и более. Большое значение уделяется увеличению скорости выполнения арифметических опе-раций.

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

Одним из направлений дальнейшего совершенствования аналоговых МП является повышение их универсальности и гибко-сти. Поэтому вместе с повышением скорости обработки большого объема цифровых данных будут развиваться средства обеспечения развитых вычислительных процессов обработки цифровой инфор-мации за счет реализации аппаратурных блоков прерывания про-грамм и программных переходов.

**По количеству выполняемых программ** различают одно-

и многопрограммные микропроцессоры.

55

* однопрограммных МП выполняется только одна про-грамма. Переход к выполнению другой программы происходит после завершения текущей программы.
* много- или мультипрограммных МП одновременно вы-полняется несколько (обычно несколько десятков) программ. Ор-ганизация мультипрограммной работы микропроцессорных управ-ляющих систем, например, позволяет осуществить контроль за со-стоянием и управлением большим числом источников или прием-ников информации.

**По числу БИС в микропроцессорном комплекте** разли-

чают однокристальные, многокристальные и многокристальные секционные МП.

Процессоры даже самых простых ЭВМ имеют сложную функциональную структуру, содержат большое количество элек-тронных элементов и множество разветвленных связей. Реализо-вать принципиальную схему обычного процессора в виде одной или нескольких БИС практически невозможно из-за специфиче-ских особенностей БИС (ограниченность количества элементов, сложность выполнения разветвленных связей, сравнительно не-большое 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 ма-гистральную структуру, т. е. в них к единой внутренней информа-ционной магистрали подключаются все основные функциональ-ные 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 основ-ными тремя функциональными частями: операционной, управ-ляющей и интерфейсной. Сложность операционной и управляю-щей частей процессора определяется их разрядностью, системой команд и требованиями к системе прерываний; сложность интер-фейсной части - разрядностью и возможностями подключения других устройств ЭВМ (памяти, внешних устройств, датчиков и исполнительных механизмов и др.). Интерфейс процессора содер-

56

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

Однокристальные МП получаются при реализации всех ап-паратурных средств процессора в виде одной БИС или СБИС. По мере увеличения степени интеграции элементов в кристалле и числа выводов корпуса параметры однокристальных МП улучша-ются. Однако их возможности ограничены аппаратурными ресур-сами кристалла и корпуса. Поэтому более широко распространены многокристальные, а также многокристальные секционные МП.

Для получения многокристального МП необходимо провес-ти разбиение его логической структуры на функционально закон-ченные части и реализовать их в виде БИС ( СБИС). Функциональ-ная законченность БИС многокристального 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 дру-гих электронных ИС (БИС). (Типичный пример - МПК БИС серии К581).

На рис. 4,а показано функциональное разбиение структуры МП при создании трехкристального микропроцессора (пунктир-ные линии), содержащие БИС операционного (ОП), БИС управ-ляющего (УП) и БИС интерфейсного (ИП) процессоров.

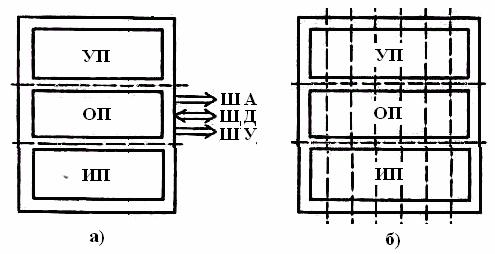


Рис. 4

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

57

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 ко-манд, а также заранее подготавливаются те данные, которые по-требуются ОП в следующих циклах работы.

Такая опережающая выборка команд экономит время ОП на ожидание операндов, необходимых для выполнения команд про-грамм. Интерфейсный процессор ИП позволяет подключить па-мять и периферийные средства к МП; по существу, является слож-ным контроллером для устройств ввода — вывода информации. БИС ИП выполняет также функции канала прямого доступа к па-мяти.

Выбираемые из памяти команды распознаются и выполня-ются каждой частью МП автономно, и поэтому может быть обес-печен режим одновременной работы всех БИС МП, т. е. конвейер-ный поточный режим исполнения последовательности команд программы (выполнение последовательности с небольшим вре-менным сдвигом). Такой режим работы значительно повышает его производительность.

Многокристальные секционные МП получаются в том слу-чае, когда в виде БИС реализуются части (секции) логической структуры процессора при функциональном разбиении ее верти-кальными плоскостями (рис.4,6). Для построения многоразрядных МП при параллельном включении секций БИС МП в них добав-ляются 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 раз-деления структуры МП на функционально и конструктивно закон-ченные 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

58

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

С момента создания однокристальные МП развились от простых специализированных 4- разрядных до 32-разрядных про-цессоров. Трехкристальные МП имеют разрядность до 32 бит и параметры , сравнимые с параметрами старших моделей рядов ми-ни-ЭВМ и средних ЭВМ общего применения.

Многокристальные секционные МП имеют разрядность от 2—4 до 8—16 бит и позволяют создавать разнообразные высоко-производительные процессоры ЭВМ.

Однокристальные и трехкристальные БИС МП, как прави-ло, изготовляют на основе микроэлектронных технологий унипо-лярных полупроводниковых приборов, а многокристальные сек-ционные БИС МП — на основе технологии биполярных полупро-водниковых приборов.

Использование многокристальных микропроцессорных вы-сокоскоростных биполярных БИС, имеющих функциональную за-конченность при малой физической разрядности обрабатываемых данных и монтируемых в корпус с большим числом выводов, по-зволяет организовать разветвление связи в процессоре, а также осуществить конвейерные принципы обработки информации для повышения его производительности.

**По структурному признаку** различают МП с фиксирован-

ной разрядностью и МП с наращиваемой разрядностью (секцион-ные МП).

МП с фиксированной разрядностью имеют строго опреде-ленную разрядность обрабатываемых слов, величина которой оп-ределяется 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 схем-но (МП с фиксированным набором команд),

59

- МП с алгоритмом управления , реализуемым программ-ным путем в виде последовательности микроопераций (МП с мик-ропрограммным управлением). Здесь система команд определена не жестко, а зависит от микропрограммы, записанной в ПЗУ, вхо-дящей в состав устройства управления. Использование микропро-граммного управления дает возможность получить необходимый набор команд, например, для воспроизведения (эмуляции) набора команд другого МП.

**По разрядности обрабатываемой информации** МП могутбыть 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 де-

лятся на синхронные и асинхронные.

* синхронных МП начало и конец выполнения каждой операции задаются устройством управления, то есть фаза начала и конца выполнения команды строго привязана к временной оси.
* асинхронных МП начало выполнения следующей опера-ции начинается сразу же после окончания выполнения предыду-щей операции**.**

**По количеству одновременно выполняемых программ**

различают одно- и многопрограммные МП.

* однопрограммных МП на текущий момент времени вы-полняется только одна программа. Переход к выполнению другой программы происходит либо по завершению этой программы, либо по специальной команде условного или безусловного пере-хода, либо по прерыванию.
* многопрограммных МП одновременно может выполнять-ся несколько программ, то есть обеспечивается мультипрограмм-ный режим работы системы.

**По виду технологии изготовления** разрабатываются и вы-

пускаются БИС МП:

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

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

60

(ТТЛШ);

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

Вид технологии изготовления БИС во многом определяет степень интеграции микросхем, быстродействие, энергопотребле-ние, 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 вы-соким энергопотреблением. МП на основе И2Л технологии обла-дают 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 тех-нология обеспечивает МП наиболее низкую стоимость, но его бы-стродействие при этом является также наиболее низким.

В процессе развития микропроцессорных средств, кроме микропроцессорных БИС, были разработаны различные инте-гральные микросхемы, выполняющие различные функции и по-зволяющие в совокупности построить микроЭВМ требуемой структуры. Эти микросхемы совместно с МП БИС образуют мик-ропроцессорный комплект (МПК БИС), который может быть оп-ределен как совокупность конструктивно и электрически совмес-тимых 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 инте-гральных микросхем, который предназначен для построения МПС и может состоять из БИС однокристального или из нескольких корпусов многокристального МП. Базовый комплект, как правило, дополняется другими типами интегральных схем, на которых реа-лизуются 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.

61

Минимальный набор микросхем из состава МПК БИС, по-зволяющих построить конкретный тип вычислительного устройст-ва, называется микропроцессорным набором интегральных схем.

* + 1. **Особенности архитектуры 32-разрядных МП** 
       1. **Микропроцессоры с RISC – архитектурой**
  + развитии архитектур МП наблюдается два подхода. Пер-вый из них относится к более ранним моделям процессоров и но-

сит название МП с CISC (Complete Instruction Set Computer) архи-

тектурой - процессоры с полным набором инструкций. К ним от-носится семейство процессоров 80 × 86. Состав и назначение их регистров существенно неоднородны, широкий набор команд ус-ложняет декодирование инструкций, на что расходуются аппарат-ные ресурсы. Возрастает число тактов, необходимое для выполне-ния инструкций.

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

* + процессорах семейства 80 × 86, начиная с i80486, приме-няется 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 воплоти-лись три основных принципа:

- ориентация системы на поддержку языка высокого уровня

* помощью развитого компилятора;
* использование примитивного набора инструкций, кото-рый полностью реализуется аппаратными средствами;
* организация памяти и ввода—вывода, которая позволяет выполнять процессором большинство инструкций за один такт.

62

Первые микропроцессоры с архитектурой RISC были разработаны и изготовлены в начале 80-х годов в Калифорнийском (г. Беркли) и Стандфордском университетах. Разработчики этих МП ставили перед собой задачу достижения наивысшей произво-дительности при наименьшей сложности. В ходе ее решения сло-жились 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 , создан-ная на этой основе, была впервые реализована в МП RISC I, разра-ботанном в г. Беркли (берклийская архитектура).

Другой подход заключается в устранении задержек конвей-ера за счет переупорядочения инструкций и интенсивного исполь-зования регистров МП при помощи оптимизирующего компилято-ра. 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 исключи-тельным характеристикам. Дело дошло до предсказаний скорого отмирания более традиционных CISC-систем. Чисто академиче-ский 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 рабо-чих станций и серверов на их базе. Более того, RISC-системы вы-шли за границы узких профессиональных приложений и находят все большее признание среди средних пользователей.

В теории цифровых логических систем есть известная ак-сиома, 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 ка-талога из десятков и сотен логических микросхем и спроектиро-

63

вать компьютер на одном типе вентиля.

Никому, кроме Сеймура Крея. Результат известен: супер-компьютер CRAY-1, созданный в рекордно короткие сроки, ока-зался меньше и быстрее всех своих предшественников. Нечто по-добное произошло и в процессе становления RISC-архитектуры. Идея, заложенная в основу RISC-архитектуры, состояла в сле-дующем: 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 поверх-ности кристалла процессора для размещения более мощных средств обработки данных. Так возникла философия RISC-архитектуры – «меньше команд - выше скорость», которая осно-вывается на двух фундаментальных постулатах:

* + скорость компьютерной обработки определяется не столь-ко быстродействием аппаратных средств, сколько хорошим взаи-модействием программного обеспечения и аппаратуры
  + за скорость всегда надо платить усложнением либо аппа-ратуры, 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 вычис-лений – «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 быстродейст-вию уступает, оптимизация откомпилированного кода теоретиче-ски позволяет в 2 - 4 раза сократить время выполнения программы

* т. д.

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

64

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

Конечно, программирование с помощью подобных насы-щенных 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 пpикладные исследования в лабораториях фирмы IBM. Было скон-струировано подмножество языка PL/1 под названием PL/8 и на-писан компилятор с оптимизацией кода для гипотетического ком-пьютера, система команд которого использовала короткие инст-рукции типа «регистр-регистр». Имитация работы этого компью-тера проводилась на мэйнфрейме IBM/370 модели 168. Этот экс-перимент дал весьма впечатляющий результат: С большинством наиболее часто употребляемых программных операторов компью-тер справилась в 2 - 3 раза быстрее, чем IBM 370/168, запрограм-мированная при помощи стандартного варианта языка PL/1.

Для последующих семейств мэйнфреймов производства IBM данное соотношение несколько уменьшилось за счет конвей-еризации процессора и большего объема кэш-памяти, однако принципиальный вывод из эксперимента в IBM не потерял своей значимости: отказ от применения редких инструкций и оптимиза-ция 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 возможно-сти для экономии оборудования без существенной потери произ-водительности. Но как только на кристалле процессора оказалось свободное место, сразу же нашлись желающие занять его под бо-лее 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. транзисто-ров), но со значительно большим быстродействием.

65

Правда, произошло это уже после того, как были сформу-лированы основные законы RISC-архитектуры. Законы RISC в са-мом начале 80-х годов почти одновременно завершились теорети-ческие исследования в области RISC-архитектуры, проводившиеся в Калифорнийском университете (г. Беркли), Станфордском уни-верситете и в корпорации IBM. Именно тогда были сформулиро-ваны четыре основных принципа RISC-архитектуры:

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

1. Все команды должны иметь одинаковую длину и исполь-зовать минимум адресных форматов; это резко упрощает логику центрального управления процессором. Другим важным следстви-ем принципа простоты адресации является то, что RISC - процес-сор способен выбирать очередную команду в темпе обработки, т. е. одну команду за один цикл.
2. Обращение к памяти происходит только при выполнении операций записи и чтения, модификация операндов в памяти воз-можна лишь с помощью команды «запись», вся обработка данных осуществляется исключительно в регистровой структуре процес-сора.
3. Система команд должна обеспечивать поддержку языков высокого уровня. Имеется в виду подбор системы команд, наибо-лее 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 за-кон RISC-архитектуры соблюдается всеми разработчиками не-укоснительно - обработка данных должна вестись только в рамках регистровой структуры и только в формате команд «регистр- ре-

66

гистр».

Регистры - основное достоинство и главная проблема RISC. Все существующие RISC-процессоры базируются на единствен-ном типе обработки данных в формате «регистр-регистр», а точ-нее, «регистр-регистр-регистр»: R1:=R2,R3. Это позволяет без су-щественных затрат времени выбрать операнды из адресуемых опе-ративных регистров и записать в регистр результат операции.

Кроме того, трехместные операции дают компилятору большую гибкость по сравнению с типовыми двухместными опе-рациями формата «регистр -память» аpхитектуpы CISC. В сочета-нии с быстродействующей арифметикой RISC-операции типа «ре-гистр-регистр» становятся очень мощным средством повышения производительности процессора. Проблема заключается в том, что в процессе выполнения задачи RISC-система неоднократно выну-ждена обновлять содержимое регистров процессора, причем за минимальное время, чтобы не вызвать длительных простоев арифметического устройства (а это прямые потери производи-тельности) . 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 модифика-ции регистров в RISC - архитектуре: аппаратный, предложенный в проектах RISC-1и RISC-2 университета в Беркли, и программный, разработанный специалистами IBM и Станфордского университе-та. Принципиальная разница между ними заключается в том, что аппаратное решение основано на стремлении уменьшить время вызова процедур за счет установки дополнительного оборудова-ния процессора, тогда как программное базируется на возможно-стях 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 содер-жит 138 регистров для хранения данных. Из них десять, именуе-мых глобальными, всегда «видны» программе; их основное назна-

67

чение - хранение данных, являющихся общими для всех процес-сов в текущем контексте программы. Остальные 128 регистров разбиты на восемь перекрывающихся окон по 22 регистра. В каж-дый момент времени программа, исполняемая на RISC-1, «наблю-дает» 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 реги-стры окна разделены на верхние, локальные и нижние. При вызове процедуры В из процедуры А активное окно регистрового файла смещается на шесть позиций так, что верхние регистры процедуры A перекрываются нижними регистрами процедуры B.

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 . Та-ким образом, вызов процедуры реализуется не сложнее, чем, ска-жем , суммирование регистровых операндов. К тому же эта опера-ция выполняется практически моментально: для обращения к про-цедуре или для возврата в точку вызова достаточно переместить указатель активного окна регистрового файла. В этом состоит важнейшая особенность архитектуры RISC-1.

Веpоятно, именно благодаря своей логической стройности архитектура RISC-1 послужила основой для разработки массовых процессоров Pyramid и SPARC, правда, с небольшими изменения-ми в организации регистрового файла Rolodex (в SPARC програм-ма «видит» окно из тех же 32 регистров, но количество глобаль-ных, верхних, локальных и нижних регистров одинаково - по во-семь в каждой зоне).

Однако структура MORS обладает двумя недостатками – оптимальное размещение процедур по окнам регистрового файла является далеко не тривиальной задачей для ОС, а выбранное чис-ло из восьми процедур, сохраняемых в регистровом файле, пред-ставляется, скорее, эмпирическим значением. Во всяком случае можно найти множество примеров, когда задача включает сущест-венно большее количество процедур и при этом возникает реаль-

68

ная проблема модификации одного или нескольких окон для ак-тивизации 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 про-цессор инициирует программу ОС, высвобождающую одно или несколько регистровых окон, т. е. передает (trap) содержимое ре-гистров в оперативную память. В случае применения RISC-1 в ка-честве 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, ес-ли прерывание происходит в момент, когда все регистровые окна заняты, то инициируется выполнение процедуры trap и время ре-акции становится недетерминированным - ситуация, крайне опас-ная 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 те-кущего 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 бо-лее трех циклов).

Именно такой разброс и является неприемлемым для сис-тем 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 включениив архитекту-ру компьютера динамического механизма быстрого сохранения регистровых окон в специальной памяти.

**5.1.2. Берклийская архитектура**

Согласно статистике, 50—70% используемых операндов

69

составляют локальные переменные и параметры процедур. Их размещение в регистровом файле МП позволяет существенно сни-зить число обращение в память. В RISC МП с берклийской архи-тектурой регистры группируются в несколько банков, чтобы для каждой процедуры процессор мог назначить свой набор регистров, переключение которого осуществляется модификацией аппаратно-го указателя.

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

Параметры Число вирт. рег. окон

Число локальных рег. в вирт. окне

Глубина перекрытия вирт. рег. окон

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | Таблица 1 |
| RISC I RISC II | | SOAR SPARC Am29000 | | |
| 6 | 8 | 8 | 7 | 8 |
| 14 | 22 | 16 | 24 | 16 |
| 4 | 6 | 8 | 8 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Число глоб. регистров | 18 | 10 | 8 | 8 | 64 |  |
| Общее число рег. в рег. | 78 | 138 | 72 | 120 | 192 |  |
| файле |  |
|  |  |  |  |  |  |

Регистры 26—31 (верхние) содержат параметры, передан-ные от вызывающей процедуры. Регистры 16—25 (локальные) ис-пользуются для хранения локальных скалярных переменных, 10— 15 (нижние) — для хранения переменных и параметров, переда-ваемых вызываемой процедуре. Регистры 0—9 предназначены для хранения глобальных переменных.

Таким образом, возможно обращение каждой процедуры к 32 регистрам. Соседние регистровые банки, используемые вызы-вающей и вызываемой процедурами, перекрываются, так что па-раметры могут быть переданы процедуре без какого-либо переме-щения данных. На каждое обращение к процедуре назначается но-вый набор регистров R10...R31. При этом нижние регистры вызы-вающей процедуры становятся верхмими вызываемой, поскольку они совмещены физически.

Таким образом, без перемещения информации параметры,

70

хранящиеся в регистрах 10—15 вызывающей процедуры, появ-ляются 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.

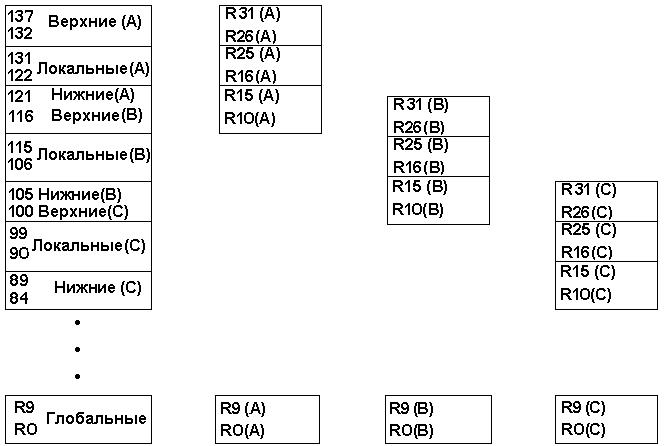


Рис. 5

Во многих программах глубина вложенности процедур превышает число регистровых окон, поэтому в МП с берклийской архитектурой освобождение регистровых банков при переполне-нии регистрового файла осуществляется путем передачи их со-держимого в память. Если глубина вложенности выходит за пре-делы логических наборов регистров, содержащихся в регистровом файле, то начинается программная или аппаратная обработка пре-рывания. Содержимое нескольких регистров передается в память, в отдельной области которой организован стек переполнения реги-стров. Сигналы переполнения/недополнения перемещают указа-тель стека на его вершину.

Эффективность такой организации обращения к процеду-рам зависит от частоты появления переполнений/недополнений, в

71

большей степени связанной с логическими изменениями глуби-ны стека, нежели с его абсолютной глубиной.

Исследования показали, что при восьми регистровых бан-ках переполнения/недополнения возникают менее чем в 1% обра-щений к процедурам. Для того, чтобы к переменным в регистрах можно было обращаться с помощью указателей , все регистры ото-бражаются в адресное пространство регулярной памяти.

**5.1.3. Станфордская архитектура**

Скорость обработки инструкций в МП с конвейерной ар-хитектурой существенно снижается из-за возникновения кон-фликтных ситуаций следующего типа:

* программа осуществляет переход, для которого требуют-ся очистка конвейера и загрузка его новыми инструкциями (зави-симость по адресу);
* инструкции запрашивают информацию, которая еще не получена от обрабатываемых в конвейере инструкций (зависи-мость по данным);
* инструкциям в конвейере одновременно требуются обра-щения к одному и тому же ресурсу — шине памяти, регистру или АЛУ.

Возможны различные случаи задержки конвейера, содер-жащего пять ступеней обработки инструкции (рис. 6): выборку ин-струкции (IF), дешифрацию инструкции (ID), выборку операнда (OF), вычисление (ОЕ) и запоминание результата (OS). В первом случае (см. рис. 6,а) выполняется инструкция безусловного пере-хода JMP, что задерживает конвейер до тех пор, пока не завершит-ся ее обработка, включающая четыре этапа:

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

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

72

ся два такта.

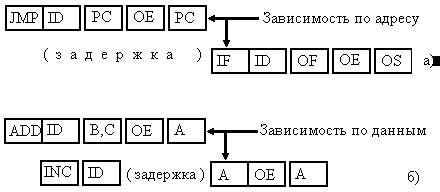


Рис. 6

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

Станфордская архитектура предусматривает устранение за-держек 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. За-держанный переход выполняется так, что инструкция, следующая за инструкцией перехода, выполняется до передачи управления в точку перехода. В это время процессор имеет возможность вы-брать инструкцию по адресу перехода и загрузить ее в конвейер.

Традиционные МП реализуют этот фрагмент программы как обычный переход - инструкции выполняются в последователь-ности : 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 ста-нет такой: 100, 101, 102, 103, 106 и т.д.; обрабатывается она опти-

73

мизирующим компилятором, который по возможности переупо-рядочивает ее так чтобы максимально использовать цикл после за-держанного перехода. Последовательность исполнения инструк-ций принимает вид : 100, 101, 102, 105 и т.д. Поскольку инструк-ция, 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 | Задержанный | Оптимиз. задержан- |  |
| переход | переход | ный 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 слу-чаев для выполнения инструкций обращения в память за один такт. Непосредственный способ выполнения этих инструкций тре-бует двух тактов: один необходим для вычисления адреса, второй

— собственно для обращения в память. Модифицированные инст-рукции обращения в память называются задержанными загрузками и предполагают некоторые изменения аппаратуры: память и реги-стры МП должны быть оснащены дополнительными портами. За-висимость 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-

74

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 по-лучили примерно одинаковое развитие и распространение.

Первые МП с архитектурой RISC (RISC I и RISС II) были разработаны и изготовлены в начале 80-х годов в Калифорний-ском университете (г. Беркли). Они имели небольшой набор инст-рукций и простые режимы адресации. Все арифметические и ло-гические инструкции выполнялись над регистровыми операндами, и только две — LOAD и STORE — использовались для обращения в память. На площади кристалла, освободившейся от схемы де-шифрации сложных команд и формирователя последовательности микрокоманд, присущих традиционной архитектуре МП, разме-щен большой регистровый файл. Организация его в виде перекры-вающихся регистровых окон позволяла каждой процедуре назна-чать новый набор регистров и передавать данные от одной проце-дуры к другой простым изменением аппаратного указателя. Реги-стровые окна, впервые примененные в берклийских МП обеспечи-вают эффективную поддержку вложенных процедур, сокращают число обращений в память и значительно повышают производи-тельность МП.

В 1983 г. в Станфордском университете был разработан и из-

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

75

Архитектура МП SOAR (Smalltalk On A RISC) является развитием берклийской ветви RISC. Он предназначен для под-держки языка высокого уровня Smalltalk-80 и функционирования в составе рабочей станции Sun. SOAR поддерживает два типа дан-ных: 31-разрядные переменные со знаком и 28-разрядные указате-ли. Каждое слово данных содержит признак типа (тег), который проверяется одновременно с выполнением арифметических опе-раций. Если оба операнда оказываются целыми, то через один такт получается правильный результат. В противном случае SOAR об-ращается к программам обработки указателей. Другой особенно-стью его является использование инструкций групповой загруз-ки—хранения, позволяющих передавать содержимое восьми реги-стров 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 инструк-ций, имеющих одинаковый 32-разрядный формат. Для обеспече-ния быстрой реакции на прерывания разработчики ARM исключи-ли из его набора команд длинные операции, которые невозможно прервать. Как и МП SOAR, ARM поддерживает инструкции груп-повой загрузки—хранения, позволяющие повысить скорость об-мена между МП и памятью. В отличие от других RISC МП, в ARM не используются задержанные переходы. Для устранения за-держек конвейера при выполнении переходов в каждую инструк-цию введено 4-разрядное поле кода условия; выполняется оно только в том случае, если установлен код соответствующего усло-вия.

МП ROMP (Research/Office division Microprocessor) - микро-

процессор отделения по автоматизации научных и Учрежденче-ских работ) корпорации IBM. Он предназначен для работы в каче-стве центрального процессора в персональном компьютере RT PC (RISC Technology Personal Computer), реализует станфордскую ар-

хитектуру и оснащен развитым оптимизирующим компилятором. Особенностью ROМР является длинное 40 - разрядное ад-ресное слово и соответствующее ему адресное пространство 1T

76

байт.

При разработке ROMP большое внимание уделялось по-вышению скорости обмена между процессором и памятью. Для этого используется аппаратно управляемый буфер преобразован-ных адресов, применявшийся в ранних моделях IBM. Он выполня-ет эффективное преобразование виртуальных адресов в физиче-ские . Кроме того, каждое слово данных в RОМР снабжается те-гом, по которому определяется получатель данных, поэтому лю-бой элемент данных может перебраться, как только он будет го-тов.

Центральный процессорный элемент (ЦПЭ) R2000 фирмы Mips Computer Systems составляет основу для построения супер-мини компьютеров этой фирмы. Он имеет станфордскую архитек-туру, 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 преобразо-ванных адресов. R2000 работает в составе трехкристального набо-ра, в который входят также сопроцессор для операций с плаваю-щей 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 перепол-нения и системных обращений. Для передачи инструкций и дан-ных используются три двунаправленные 32-разрядные шины, ка-ждая линия которых заряжается с помощью большого n-канального транзистора. Высокая производительность ЦПЭ обес-печивается за счет сопроцессора для операций с плавающей точ-кой и устройства управления памятью, входящих в МП набор.

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

77

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

МП Аm29000 фирмы AMD отличается наивысшей произво-дительностью среди серийных RISC МП и реализует усовершен-ствованную версию берклийской архитектуры. В него входит ре-гистровый файл емкостью 192 регистра, разделенный на банки по 16 регистров; использует он задержанные переходы для устране-ния 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 Аm29000 вве-ден буфер преобразованных адресов.

Трехкристальный МП набор Clipper С100 корпорации Fairchild позволяет реализовать суперЭВМ на одной плате. Архи-тектура Clipper С100 имеет некоторые черты RISC, например ап-паратную 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 дан-ных и девять режимов адресации, а его система прерываний вклю-чает 256 векторных прерываний и 128 системных обращений.

Набор инструкций Clipper С100 содержит 101 аппаратно-реализованную инструкцию и 67 макроинструкций, выполняющих преобразование чисел из формата с фиксированной точкой в фор-мат 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. Вычис-лительный модуль Clipper С100 работает в программной среде ОС Unix и языков высокого уровня Си, Фортрая и Паскаль.

МП система МС88000 фирмы Motorola включает процессор МС88100 и два кристалла памяти МС88200.

МС88100 содержит регистровый файл небольшой емкости, блоки обработки чисел в формате с фиксированной и плавающей точкой, а также до шести заказных блоков специальных функций.

78

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 повы-сило 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 аль-тернативной архитектуры, нацеленной, прежде всего, на снижение нерегулярности потока команд уменьшением их общего количест-ва. Это было реализовано в RISC-процессорах.

Сокращение нерегулярности потока команд позволило обо-гатить топологию RISC-процессоров пространственным паралле-лизмом, специализированными аппаратными АЛУ, независимыми кэш данных и команд, раздельными шинами ввода -вывода. По-следние, в частности, увеличили длину конвейеров команд. Всё это повысило и производительность - увеличением числа опера-ций, 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 ос-новном, клонами процессоров Intel серии x86, производимыми

AMD, Cyrix, а RISC -чипами Alpha, PowerPC, SPARC. Уступая во многом последним, процессоры x86 сохранили лидерство на рын-ке персональных систем лишь благодаря совместимостис про-

79

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

Несмотря на формальное разделение «сфер влияния», меж-ду представителями этих архитектур в начале 90-х годов началась острая конкуренция за улучшение характеристик. В первую оче-редь, производительности и её отношения к трудоёмкости разра-ботки процессоров.

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

Для устранения недостатка в Pentium реализованы раздель-ные кэш для команд и данных, позволяющие после переходов пе-резагружать лишь команды- такое решение называется Гарвард-ской архитектурой, а также предсказание переходов, снижающее частоту перезагрузок. Последнее достигается предварительной за-грузкой в кэш команд с обоих разветвлений . Введение второго це-лочисленного тракта, состоящего из АЛУ, адресного блока, шин данных/команд, и работающего на общий блок регистров, повы-сило производительность поддержкой параллельной обработки целочисленных данных . Развитием данной тенденции стало обо-гащение 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 последователь-но, что снижает эффективность работы процессора. Частично по-правило ситуацию создание оптимизирующих рекомпиляторов, например, Pen-Opt фирмы Intel, разделяющих по возможности та-кие команды.

Реализация описанного управления обработкой команд

80

CISC-формата вызвала дополнительный рост трудоёмкости раз-работки Pentium в сравнении с i8086/i80486, что привело не только к увеличению её реального срока на 27% в сравнении с ожидае-мым, но и к проявлению ошибок в первых моделях данного про-цессора

Учтя это, компания Intel реализовала в Pentium Pro RISC-подобную организацию вычислений. Интерпретация команд 80×86 внутренними - RISC86 - инструкциями VLIW-формата по-мимо снижения нерегулярности их потока, обеспечила синхрон-ную загрузку четырёх операционных - по два с плавающей и фик-сированной запятой - АЛУ этого чипа. Термин VLIW расшифро-вывается 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 раздельными ши-нами «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, инте-грация кэш второго уровня позволила умножать частоту ядра в 5-6 раз.

В архитектуре Р6 RISC - решения впервые в семействе 80×86 перестали быть лишь дополнением исконных CISC - средств повышения производительности - роста разрядности, от-ложенной записи шины и других. Поэтому частица PRO в назва-нии первого процессора этой серии обозначает «полноценная

RISC-архитектура» (Precision RISC Organization).

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

81

стик этих процессоров, к их цене.

Снижение трудоёмкости и длительности разработки аппа-ратно – программных реализаций алгоритмов работы Pentium Pro, позволило достигнуть роста характеристик сочетанием преиму-ществ RISC&CISC архитектур.

Сказанное иллюстрирует и организация современных RISC-процессоров. Их отличает, в данном случае, развитие систем команд с целью сохранения иерархической совместимости и сни-жения трудоёмкости разработки программ. Это сближает техноло-гии обработки команд процессорами упомянутых архитектур. На-пример, SuperSparc взяли от последних моделей 80×86 предсказа-ние переходов и предварительную интерпретацию кода.

Таким образом, развиваясь, каждая из рассмотренных архи-тектур, «отказавшись» от своих черт - CISC от скалярности вы-числений, RISC от «простоты» системы команд, приобрела луч-шие характеристики её представителей. Это подтверждает и про-

цессор Merced, разработанный Intel и Hewlett Packard.

Имеющиеся сведения позволяют предположить, что его ар-хитектура продолжит тенденции Pentium Pro по оптимизации об-работки внутренних VLIW – подобных команд реализацией эф-фективных архитектурных решений при одновременной оптими-зации 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 улуч-шили свои характеристики. Это позволило предположить, что дальнейшее развитие массовых процессоров пройдёт по пути раз-вития топологических и микропрограммных решений вычисли-тельного ядра RISC - организации при одновременном повышении возможностей CISC - подобной системы команд.

1. **МП с традиционной архитектурой** 
   1. **Intel Pentium 4**

82

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 кри-сталле процессора общую кэш-память 2-го уровня L2 (рис. 7).

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

Архитектура МП является суперскалярной, что обеспечива-ет одновременное выполнение нескольких команд в параллельно работающих исполнительных устройствах. Суперскалярность ар-хитектуры реализуется путем организации исполнительного ядра процессора в виде ряда параллельно работающих блоков. Арифме-тико-логические блоки ALU производят обработку целочисленных операндов, которые поступают из заданных регистров блока реги-стров замещения (БРЗ). В эти же регистры заносится и результат операции. При этом проверяются также условия ветвления для ко-манд условных переходов и выдаются сигналы перезагрузки кон-вейера команд в случае неправильно предсказанного ветвления. Исполнительное ядро работает с повышенной скоростью выпол-нения операций.

Адреса операндов, выбираемых из памяти, вычисляются блоком формирования адреса (БФА), который реализует интер-фейс 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 ре-гистр БРЗ и 24 адреса для записи из регистра в память . При этом БФА формирует адреса операндов для команд, которые ещё не по-ступили 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

83

образом реализуется процедура предварительного чтения дан-ных для последующей их обработки в исполнительных блоках, ко-торая называется спекулятивной выборкой.

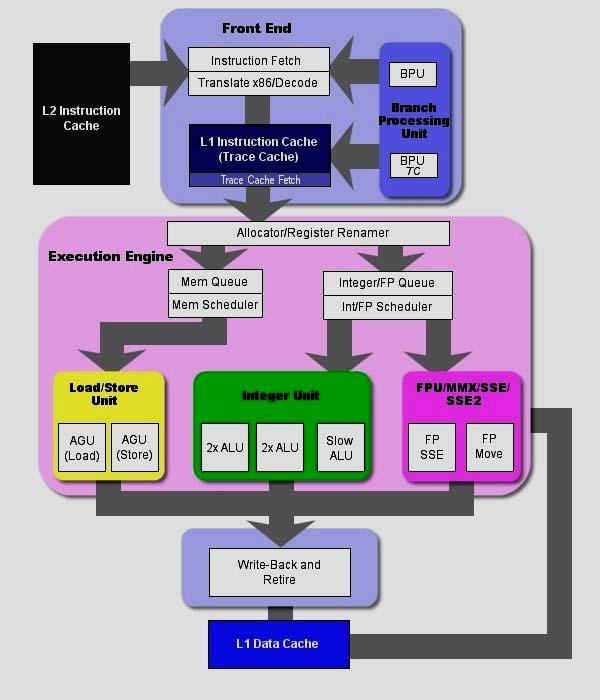


Рис. 7 Аналогичным образом организуется параллельная работа

блоков SSE, FPU, MMX, которые используют отдельный набор ре-гистров и блок формирования адресов операндов.

84

Хотя конвейер Pentium 4 является намного более длин-ным, он выполняет те же функции что и конвейеры других про-цессоров.

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

Особое внимание стоит уделить тому, что кэш -память L1 разделена и кэш инструкций находится фактически на препроцес-соре . Он называется отслеживающим кэшем (Trace Cache) и явля-ется одной из важных инноваций в Pentium 4. Эта кэш -память ока-зывает сильное влияние и на конвейер, и на основной поток инст-рукций.

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 час-ти, более единообразные, с которыми проще работать, - микроко-манды. Фактически, эти инструкции применяются при внеочеред-ном 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 - не-сколько ступеней, декодирование - несколько, и т.д.).

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

85

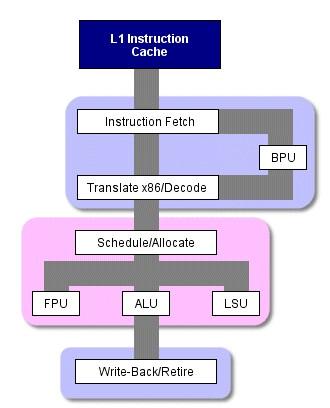


Рис. 8

Кэш инструкций в Pentium 4 принимает транслированные и декодированные микрокоманды, готовые к передаче на внеоче-редное выполнение, и формирует из них мини-программы , назо-вем их трейс -последовательностями (traces). Именно эти мини-программы (а не 80x86-код, созданный компилятором) и выполня-ет 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 мик-рокоманд 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-

86

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

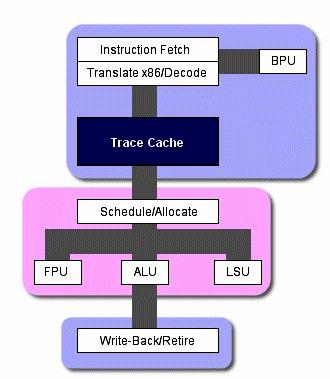


Рис. 9

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

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

87

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 пред-полагается к выполнению при предсказании ветвления. Поэтому если имеется фрагмент x86-кода с ветвлением, отслеживающий кэш построит трейс -последовательность из инструкций до ветвле-ния, 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 ветв-ления (BPU) должно определить, какую из ветвей нужно спекуля-тивно 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 приво-дит к появлению нежелательного пузырька (пустая ячейка в кон-вейере - pipeline bubble). В случае же использования отслеживаю-щего кэша, код после ветвления уже готов к выполнению сразу же после инструкции ветвления, поэтому показанных задержек не возникает.

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

88

строки остается пустой. Если инструкция ветвления находится вначале строки кэш-памяти L1, то в считанной строчке будет на-ходиться только одна эта инструкция . При использовании отсле-живающего кэша считанные строчки могут содержать как инст-рукции ветвления, так и спекулятивный код после них. Таким об-разом, в 6-командных строчках не возникает потерянного места.

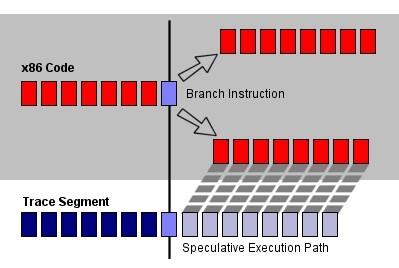


Рис.10.

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

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

89

ходует множество транзисторов на усиленный блок декодирова-ния 80x86-макрокоманд, что позволяет за цикл декодировать дос-таточно много громоздких 80x86-инструкций в микрокоманды для загрузки исполнительного модуля. В случае же с Pentium 4 нали-чие отслеживающего кэша означает, что большая часть кода заби-рается из него уже в виде готовых микрокоманд , так что здесь от-падает надобность в трансляторах и декодерах с высокой пропуск-ной 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 пропуск-ной способности декодера Athlon, но отслеживающий кэш в Pentium 4 позволяет ему достичь или даже обойти производитель-ность 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 , инструк-ции по строковой обработке. Как и в Athlon, в Pentium 4 существу-ет специальное ПЗУ микрокода, которое обрабатывает эти гро-моздкие инструкции, что позволяет разгрузить аппаратный деко-дер 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, разработ-чики поступили следующим образом: как только при создании от-слеживаемого 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, содержа-щее последовательность микрокоманд данной инструкции. Позд-нее, в режиме выполнения, когда отслеживающий кэш будет пере-давать поток инструкций на ступень выполнения, при попадании на такую метку он временно приостановит работу и на время пе-

90

редаст управление потоком инструкций ПЗУ микрокода . Здесь уже ПЗУ будет выдавать в поток инструкций требуемую последо-вательность микрокоманд (как определено меткой). После этого, оно возвратит управление обратно, и отслеживающий кэш про-должит передавать инструкции. Исполнительному модулю безраз-лично, откуда поступает поток инструкций (из отслеживающего кэша или из ПЗУ). Для него все это выглядит как непрерывный по-ток 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 ина-че, нежели стандартный кэш инструкций L1, то для того, чтобы оценить, как его размер влияет на производительность всей систе-мы, 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 про-цессора и обращения к памяти и внешним устройствам (организа-ция памяти и реализация обмена по системной шине), средства об-работки прерываний и исключений.

* процессоре Pentium 4 реализуется архитектура IA-32 (Intel Architеcture-32), общая для всех 32-разрядных микропроцес-

соров Intel, начиная с 80386. Модели Pentium II Xeon и Pentium III Xeon ориентированы на работу в высокопроизводительных муль-типроцессорных системах (серверах, рабочих станциях). Для этих же приложений в 2001 году выпущена модификация процессора Pentium 4 с поддержкой мультипроцессорного режима работы (на ядре Foster).

* процессе развития IA-32 производилось расширение воз-можностей обработки данных, представленных в различных фор-матах (рис. 11). Процессоры 80386 выполняли обработку только целочисленных операндов. Для обработки чисел с плавающей точ-

91

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 обес-печивается 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 пла-вающей точкой.

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

92

"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 процессо-ра, используемых для промежуточного хранения данных (рис. 12). Кроме 32-разрядных регистров для хранения целочисленных опе-рандов, процессоры Pentium содержат 80-разрядные регистры, ко-торые 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 ис-пользуются как 64-разрядные регистры MM0-MM7, где могут хра-ниться несколько операндов (8 8-разрядных, 4 16-разрядных, 2 32-разрядных или один 64-разрядный), над которыми одновременно выполняется поступившая в процессор команда (арифметическая, логическая, сдвиг и т.д.).

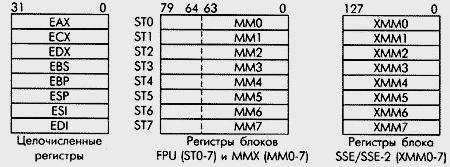


Рис.12

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

93

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 об-рабатывать целочисленные операнды: 16 8-разрядных, 8 16-разрядных, 4 32- разрядных или 2 64- разрядных. В результате про-изводительность процессора Pentium 4 при выполнении таких опе-раций оказывается вдвое выше, чем Pentium III.

Операции SSE2 позволяют существенно повысить эффек-тивность процессора при реализации трехмерной графики и ин-тернет-приложений, обеспечении сжатия и кодирования аудио- и видео-данных. Что касается базового набора команд и используе-мых способов адресации операндов, то они практически полно-стью совпадают с набором команд и способов адресации в преды-дущих моделях Pentium. Процессор обеспечивает реальный и за-щищённый режимы работы, реализует сегментную и страничную организации памяти. Таким образом, пользователь имеет дело с хорошо знакомым набором регистров и способов адресации, мо-жет работать с базовой системой команд и известными ва-риантами реализации прерываний и исключений, которые харак-терны для всех моделей семейства Pentium.

Pentium 4 является первым IA-32 (32-bit Intel Architecture)

процессором, использующим не P6 архитектуру. Сегодня эта ар-хитектура 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 P6 конвейера, которым оснащен Pentium III и в четыре раза длиннее, чем P5 кон-вейер. 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 бу-дет дебютировать на скорости 1.4 ГГц и выше. Но 20-ти шаговый конвейер Pentium 4 приводит к уменьшению значения IPC (инст-рукций за такт).

94

Имеется множество путей восполнения низкого IPC. Один из них, наиболее очевидный, заключается в простом увели-чении 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 значи-тельно больший результат, потому что выполняет больше инст-рукций за такт, чем Pentium 4.

В современных процессорах предусмотрены средства уве-личения эффективности конвейеров засчет предсказания хода программы. Когда процессор правильно предсказывает следую-щую 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 мень-ший штраф за неправильный переход, чем процессор с 20 шаго-вым конвейером.

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

Как уже указывалось ранее, АЛУ Pentium 4 работает на уд-военной 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 ос-новной 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 посред-ник между стадией декодирования и первой стадией выполнения. Trace cache по существу кэширует декодированные micro-ops (ин-струкции 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 коман-ды Pentium 4 мог обратиться к trace cache, получить декодирован-ные micro-ops и начинать выполнение.

95

Это помогает уменьшать штраф, связанный с неправиль - но 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 пред-ставлены в порядке выполнения. Это добавляет некоторый потен-циал для неправильного предсказания пути выполнения кэшируе-мых micro-ops, однако Intel уверена, что это будет компенсиро-ваться 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 дос-тижения 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 ско-рость передачи 24GB/s. Pentium 4 на той же тактовой частоте име-

ет 48GB/s. L1 в Pentium 4 (включая Execution Trace Cache) дубли-

руется в L2.

**5.2.2. AMD Athlon**

Микроархитектура Athlon представлена на рис.13. С целью увеличения пропускной способности декодеров, которые переко-дируют 80х86-команды во внутренние макрооперации, в блок кэш-памяти команд первого уровня добавлена специальная кэш-память предварительного декодирования. Эти макрооперации представ-ляют собой по сути RISC-команды, которые, собственно, и испол-

96

няются. Подобная схема перекодирования из 80х86 в RISC ис-пользуется и в Pentium III, и в предыдущих процессорах AMD, и в некоторых других 80х86-совместимых процессорах.

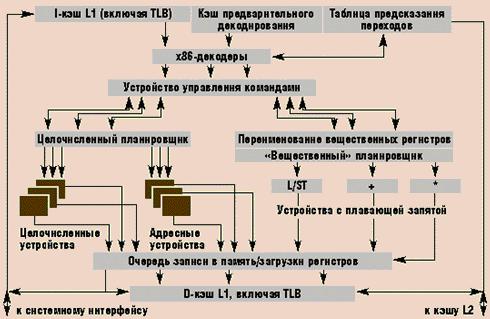


Рис .13

В Athlon таких декодеров три, а число выдаваемых на вы-полнение за такт команд больше, чем в Pentium III. Команды в де-кодеры 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: перво-го уровня — на 24 строки, а второго — на 256.

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

97

в блок управления командами, емкость которого 72 строки. Athlon является суперскалярным микропроцессором с вне-

очередным спекулятивным выполнением команд. Большая ем-кость очереди команд позволяет Athlon эффективно использовать свои ресурсы: число функциональных исполнительных устройств в нем больше, чем в Pentium III.

Совокупность вышеуказанных факторов уже объясняет, по-чему производительность Athlon выше, чем в Pentium III. В Athlon имеется три целочисленных устройства и три устройства с пла-вающей запятой. Кроме того, Athlon содержит три адресных уст-ройства. 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 уст-ройств) строк соответственно.

Следует отметить , что в составе МП имеются следующие устройства с плавающей запятой: cумматор, умножитель и модуль загрузки регистров/записи в память. Они могут работать парал-лельно, 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 час-тоте. Кроме того, умножитель Athlon - это настоящий конвейер. Физических регистров с плавающей запятой в Athlon - 88, что по-зволяет 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 ве-щественном конвейере) позволяет легче поднимать тактовую час-тоту. С другой стороны, это вызывает проблему заполнения кон-вейеров: 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

98

45, из них 24 - новых, в том числе: 12 команд целочисленной ма-тематики для обработки видео и распознавания речи; 7 команд пе-ресылки данных, ориентированных на программы, подобные Inter-net-приложениям, работающим с графическими данными; 5 новых команд для цифровой обработки сигналов

Кроме рассмотренных выше блоков микропроцессора, Ath-lon имеет встроенное управление внешней кэш-памятью второго уровня и сопряжение с системной шиной. Интеграция в микро-процессор функций управления внешней кэш-памятью позволяет Athlon иметь L2, расположенный на выделенной шине с програм-мируемой частотой. Поддерживается совместимость с индустри-альными стандартами SRAM, в том числе DDR и SDR. Кроме то-го, 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 пре-имуществом коммутаторов является то, что в отличие от систем-ной шины они не имеют конфликтов и обеспечивают гарантиро-ванный 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 высо-кую пропускную способность, чем у шины Pentium III. Однако, по некоторым данным, на тестах пропускной способности оператив-ной памяти (STREAM) Athlon лишь незначительно опережает Pen-tium III. Важным преимуществом системного интерфейса Athlon является расщепленная обработка транзакций (до 24 на процессор против 4 в Pentium III). При этом пакетный протокол может пере-давать блоки в 64 байт против 32 байт у Pentium III.

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

99

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

Athlon имеет площадь 128 кв. мм и производится по 0,18-микронной технологии с шестислойной металлизацией. Он ис-пользует 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 свя-заны 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, содержа-щего 1.3 миллиона вентилей. Центральной частью этой архитекту-ры является шина операндов (в реализации это шесть 80-битовых шин), соединяющая регистровые файлы и исполнительные уст-ройства.

Процессор имеет 10 исполнительных устройств, которые работают одновременно и независимо, и два регистровых файла. Файл регистров общего назначения имеет 32- битовую организа-цию. 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 . Внеш-няя шина имеет также специальные сигналы управления, обеспе-чивающие 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.

100

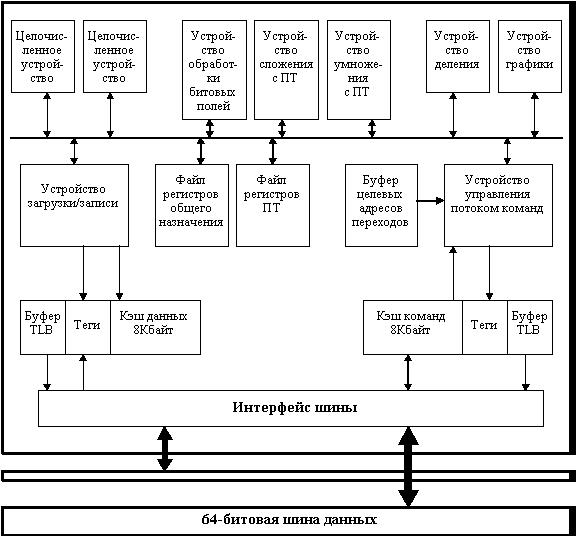


Рис. 14

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

101

Суперскалярная архитектура процессора базируются на реализации возможности завершения команд не в порядке их по-ступления 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 ста-рые значения регистров при выполнении и завершении операций не в предписанном программой порядке, и позволяет аппаратно восстановить необходимое состояние в случае прерывания.

* процессоре предусмотрено несколько способов ускоре-ния обработки условных переходов. Один из них, предсказание направления перехода, позволяет компилятору сообщить процес-сору предпочтительное направление перехода.

Для выполняемых переходов используется буфер целевых адресов перехода емкостью 32 строки, позволяющий быстро вы-брать две команды по целевому адресу перехода. Механизм пред-сказания направления переходов позволяет одновременно выпол-нять эти команды и оценивать условие перехода. Для предсказан-ного направления перехода разрешено спекулятивное (условное) выполнение команд. Если направление перехода предсказано не-верно, исходное состояние процессора восстанавливается с помо-щью буфера истории. Выполнение программы в этом случае будет продолжено с "правильной" команды.

* каждом такте процессор может выдавать на выполнение две команды. В большинстве случаев выдача команд осуществля-ется в порядке, предписанном программой. Команды записи и ус-ловных переходов могут посылаться на буферные станции резер-вирования, из которых они в дальнейшем будут выданы на выпол-нение. Команды загрузки могут накапливаться в очереди. Таким образом эти команды не блокируют выдачу второй команды из па-ры.

Большое количество исполнительных устройств позволяет осуществлять одновременную выдачу двух команд во многих си-туациях: 2 целочисленные команды, 2 команды с плавающей точ-кой, 2 графические команды или любая комбинация перечислен-ных команд.

102

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

При построении многопроцессорной системы все процес-соры и основная память размещаются на одной плате. Для обеспе-чения хорошей производительности системы каждый процессор в такой конфигурации снабжается кэш-памятью второго уровня ем-костью 256 Кбайт. Протокол поддержания когерентного состояния кэш- памяти (протокол наблюдения) базируется на методике запи-си с аннулированием, гарантирующей размещение модифициро-ванной 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 недейст-вительными 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 , отра-жающей доминирующее положение компании на рынке UNIX-рабочих станций и серверов. Процессоры с архитектурой SPARC лицензированы и изготавливаются по спецификациям Sun не-сколькими производителями, среди которых следует отметить компании Texas Instruments, Fujitsu, LSI Logic, Bipolar International Technology, Philips и Cypress Semiconductor. Эти компании осуще-

ствляют поставки процессоров SPARC не только самой Sun Microsystems, но и другим известным производителям вычисли-

103

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

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

В отличие от большинства RISC архитектур, SPARC ис-пользует регистровые окна , которые обеспечивают удобный меха-низм передачи параметров между программами и возврата резуль-татов. Архитектура SPARC была первой коммерческой разработ-кой, реализующей механизмы отложенных переходов и аннулиро-вания команд. Это давало компилятору большую свободу запол-нения времени выполнения команд перехода командой, которая выполняется в случае выполнения условий перехода и игнориру-ется 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, объявлен-ная осенью 1987 года (до этого времени компания Sun использова-ла 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 разработ-ки для реализации своих изделий SPARCstation 1, SPARCstation 1+, SPARCstation IPC, SPARCstation ELC, SPARCstation IPX, SPARCstation 2 и серверов серий 4ХХ и 6ХХ. Тактовая частота процессоров SPARC была повышена до 40 МГц, а производитель-

ность - до 28 MIPS.

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

104

SPARCstation/SPARCserver 10 и SPARCstation/SPARCserver 20.

Имеется несколько версий этого процессора, позволяющего в за-висимости от смеси команд обрабатывать до трех команд за один машинный такт, отличающихся тактовой частотой. Процессор SuperSPARC (рис. 15) имеет сбалансированную производитель-ность на операциях с фиксированной и плавающей точкой. Он имеет внутреннюю кэш-память емкостью 36 Кб (20 Кб - кэш ко-манд и 16 Кб - кэш данных), раздельные конвейеры целочислен-ной 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 взаимозаменяе-мых процессорных модулях трех типов, отличающихся наличием и объемом кэш-памяти второго уровня и тактовой частотой. Мо-дуль 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 называе-мому 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 по-следние три модели и по одному 1 Мб внешней кэш-памяти на ка-ждый 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 существую-щих 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.

105

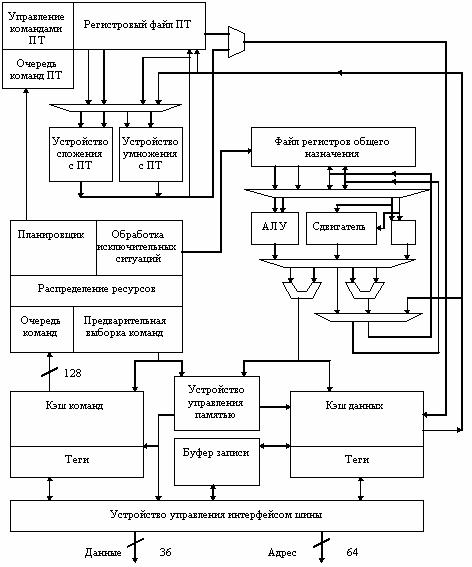


Рис. 15

Sun совместно с Fujitsu создали также новую версию кри-сталла MicroSPARC II с встроенной кэш-памятью емкостью 24 Кб. На его основе построены рабочие станции и серверы

SPARCstation/SPARCserver 4 и SPARCstation/SPARCserver 5, ра-

ботающие на частоте 70, 85 и 110 МГц.

106

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 час-тоты других архитектур процессоров. Чтобы ликвидировать это отставание, а также в ответ на появление на рынке 64-битовых процессоров компания, Sun проводила в жизнь программу модер-низации . В соответствии с этой программой Sun довела тактовую частоту процессоров MicroSPARC до 100 МГц в 1994 году ( про-цессор 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 одно-процессорные и двухпроцессорные рабочие станции с тактовой частотой процессора 100 и 125 МГц. К середине 1995 года такто-вая 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 - про-цессоры 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. Эконо-мические соображения заставляют принимать решения, основой которых является массовая технология. Системы microSPARC обеспечивают высокую производительность при умеренной такто-вой частоте путем оптимизации среднего количества команд, вы-полняемых за один такт. Это ставит вопросы эффективного управ-ления конвейером и иерархией памяти. Среднее время обращения к памяти должно сокращаться, либо должно возрастать среднее количество команд, выдаваемых для выполнения в каждом такте,

107

увеличивая производительность на основе компромиссов в кон-струкции процессора.

MicroSPARC-II (рис. 16) является представителем семейст-ва микропроцессоров SPARC. Основное его назначение - одно-процессорные низкостоимостные системы. Он представляет собой высокоинтегрированную микросхему, содержащую целочисленное устройство, устройство управления памятью, устройство плаваю-щей точки, раздельную кэш-память команд и данных , контроллер управления микросхемами динамической памяти и контроллер шины SBus.

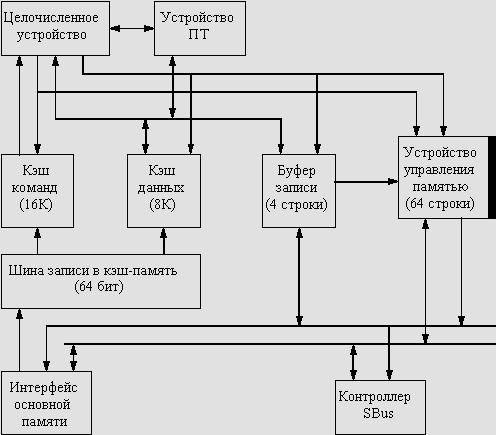


Рис. 16

108

Основными свойствами целочисленного устройства microSPARC-II являются:

* + пятиступенчатый конвейер команд;
  + предварительная обработка команд переходов;
  + поддержка потокового режима работы кэш-памяти команд
* 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. Уст-ройство плавающей точки обеспечивает выполнение операций в соответствии со стандартом IEEE 754.

Устройство управления памятью выполняет четыре основ-ных функции. Во-первых, оно обеспечивает формирование и пре-образование виртуального адреса в физический. Эта функция реа-лизуется с помощью ассоциативного буфера TLB. Кроме того, устройство управления памятью реализует механизмы защиты па-мяти. 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 емко-стью 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. Впро-чем, 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-

109

рующим составом были, естественно, единая идеология, при-

шедшая на смену CISC - EPIC (Explicitly Parallel Instruction Computing), и новая архитектура - IA-64, включающая в себя на-бор 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 разновидно-стях 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, естест-венно , процессор содержит исполнительные модули - для опера-ций с целыми числами, с числами с плавающей запятой, и т.д.

* + отличие от Pentium и его последователей, разбирающихся
* коде самостоятельно, EPIC-процессоры сильно полагаются на компилятор, который должен сам проанализировать код на пред-мет нахождения оптимальных мест для распараллеливания его вы-полнения, и снабдить процессор этой информацией. Поэтому и используется «explicitly» - процессор не должен сам пытаться по-нять, что можно исполнять параллельно, а что нет, и т.д. - все это ему уже заранее объяснит компилятор. Плюс, мощные механизмы по предсказанию переходов, предварительному выполнению кус-ков кода, предварительной загрузке данных, и тому подобные ве-щи - загрузка исполнительных блоков должна быть распределена максимально равномерно.

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

110

с памятью и четыре - для операций с плавающей точкой. Физи-ческая память адресуется 44-бит числами, что на самом деле огра-ничивает ее объем "всего лишь" 17.6 Терабайт, блоки для опера-ций с плавающей точкой работают с числами в 82-бит представле-нии.

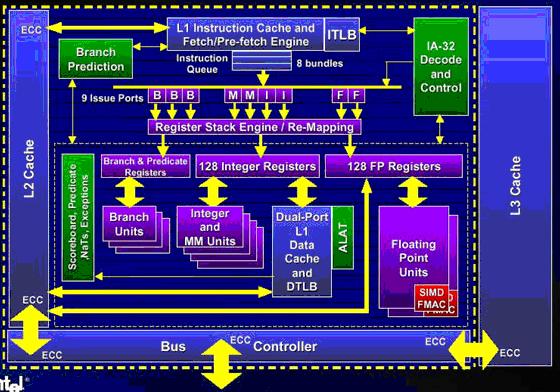


Рис.17

От идеи реализовать 32-бит 80x86 ядро в аппаратном виде Intel отказался, сочтя это слишком неэффективным использовани-ем площади кристалла. Так что для того, чтобы получить возмож-ность исполнения Itanium 80х86 кода, была создана система трансляции, которая на лету преобразует 80x86 код в IA-64.

111

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 относит-ся скорее к издержкам переходного периода . Тем не менее, факт остается фактом: это семейство для решения 32-бит задач не при-способлено. 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 ар-хитектуры . Характерный штрих - чипсет для Itanium, 460GX, под-держивает 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 способно-стью до 12.8 Гбайт/с.

Еще одной задачей Itanium было решить вопрос с компиля-торами - ведь EPIC-процессоры , как уже упоминалось, очень сильно от них зависят. В отличие от компиляторов для 80x86 про-цессоров, которые на их производительность почти не влияли, здесь компиляторы являются полноправными партнерами процес-сора - ведь они снабжают его крайне необходимой для работы ин-формацией, и от того, насколько качественной она будет, будет за-висеть 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

112

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

Дальше планируется еще большее распараллеливание мак-симально 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.

* 1. **Athlon 64 AMD**
* 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 сроч-ном порядке решила нацелить на рынок настольных систем и од-нопроцессорный 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 - раз-

рядных 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 про- | 0.13 мкм | 0.13 мкм | 0.13 мкм |  |
| изводства |  |
|  |  |  |  |
|  |  |  |  |  |
| Число транзи- | 105.9 млн. | 105.9 млн. | 54.3 млн. |  |
| сторов |  |
|  |  |  |  |
|  |  |  |  |  |
| Площадь ядра | 193 кв.мм | 193 кв.мм | 101 кв.мм |  |
|  |  |  |  |  |

113

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

\* Заметим, что память в Athlon 64 и Athlon 64 FX такту-ется относительно частоты ядра, поэтому реальные час-тоты для памяти в этом случае составляют 129.4, 157.1 и 200 МГц.

Фактически, Athlon 64 отличается от своего старшего собра-та 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 од-ноканального контроллера памяти в Athlon 64 очевиден: это более низкая теоретическая пропускная способность.

Учитывая, что Athlon 64 способен работать с DDR400 памя-тью, 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. Преиму-щество же контроллера памяти Athlon 64 заключается в том, что он, в отличие от контроллера Athlon 64 FX, поддерживает обыч-ные нерегистровые модули памяти. Такие модули по сравнению с регистровыми более дешевы, имеет более агрессивные тайминги и работают быстрее, даже при одинаковых с регистровыми модуля-ми настройках. То есть при более низкой пропускной способности, обеспечиваемой контроллером памяти Athlon 64, подсистема па-

114

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

АMD 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 се-

мейств Athlon 64 FX и Opteron.

Помимо перечисленных выше особенностей , есть в новых процессорах Athlon 64 и еще одна. Эти процессоры обладают под-держкой 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 ком-пании. 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 присталь-ное внимание понижению уровня тепловыделения своих настоль-ных процессоров.

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

115

ства Athlon XP имели возможность перехода в «ждущий ре-жим» (Halt/Stop Grant) при выполнении команды HALT, что выли-валось в понижение температуры процессора при его загрузке ни-же 100%. Однако теперь AMD пошла еще дальше. В новых про-цессорах 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 использова-нием 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 систе-мы 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 по-добных задачах мощности процессора в полной мере не использу-ются. 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 вычис-лительных 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 4x. Это, кстати, обу-славливает и тот факт, что процессоры Athlon 64 3200+ поставля-ются с незафиксированным коэффициентом умножения. В таком режиме процессор продолжает работать до тех пор, пока его за-грузка не превысит 70-80%. В частности, мы смогли запустить од-новременно дефрагментацию диска, проигрывание файлов с рас-ширением mp3 (аудифайлов) и просмотр MPEG-4 (видеофайлов)

116

ролика, в то время как процессор продолжал работать на часто-те 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+ со-ставляет 1.8 ГГц, а напряжение питания 1.4В. Достигается это вновь за счет уменьшения множителя, на этот раз до 9x. 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 часто-той тепловыделение процессора Athlon 64 3200+ резко падает. Для сравнения приведем таблицу 4 с тепловыделением этого процес-сора в основных режимах.

Таблица 4

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

Таким образом, использование технологии Cool’n’Quiet по-зволяет значительно снизить температуру процессора не только в моменты простоя, но и во время выполнения ряда задач , не тре-бующих от МП максимальной производительности. Что немало-важно, быстродействие МП в задачах, требовательных к процес-сорным ресурсам, при этом совершенно не снижается. В итоге, при применении систем охлаждения с вентиляторами с перемен-ной скоростью, использование технологии Cool’n’Quiet может по-зволить значительно снизить уровень шума.

117

**6.3. UltraSPARC III Sun**

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

UltraSPARC III представляет собой высокопроизводитель-ный суперскалярный микропроцессор, предназначенный для ис-пользования в мощных рабочих станциях и серверах. Он ориенти-рован на работу в масштабируемых микропроцессорных комплек-сах.

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 со-ответственно;
* кэш предвыборки и записи объемом по 2 КБ каждый;
* системный интерфейс;
* контроллер основной памяти;
* контроллер L2;
* таблица тэгов L2.

Функции исполнительных устройств распределяются сле-дующим образом. Целочисленные: 2 АЛУ (Integer ALU, ALU/Load), специализированное исполнительное устройство (Load/Store) и устройство обработки ветвлений (Brach Unit). Пла-вающие: устройства плавающей арифметики/графики (FP adder, FP mul/div/sqrt). Integer ALU, ALU/Load отвечают за выполнение це-

лочисленных операций. Блок Integer ALU содержит фактически два устройства, каждое из которых может выполнять все одно-тактные операции. Блок ALU/Load выполняет многотактные ко-манды (умножение, деление ) и рассчитывает адреса в командах предварительной выборки. В этот блок команды поступают на вы-полнение по одной, в то время как всего микропроцессор может

118

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

UltraSPARC III.

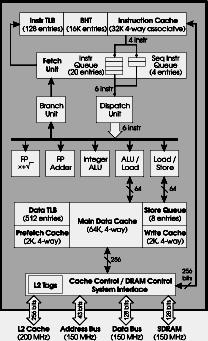


Рис. 18

В UltraSPARC III предусмотрены четырехканальный час-тично-ассоциативный D-кэш (кэш данных) первого уровня емко-стью 64 Кбайт и четырехканальный частично-ассоциативный I-

119

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 обу-словлено, по всей видимости, высокой частотой процессора (600 МГц).

Кроме кэш-памяти первого уровня, в UltraSPARC III имеет-ся 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 вы-борки поддерживает одновременное выполнение двух команд за-грузки регистров с общей задержкой всего три такта; за 14 тактов процессора из кэш второго уровня в кэш предварительной выбор-ки можно записать 64 байт данных.

В обоих кэш первого уровня содержатся также буферы бы-строй переадресации (I-TLB и D-TLB соответственно), ускоряю-щие 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 преду-

смотрено 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 четырехка-нальный 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,

120

использующего алгоритм сквозной записи, здесь применяется алгоритм обратной записи.

Применение очереди записи и соответствующего кэш спо-собствует 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 удается 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 способ-ность уменьшилась втрое.

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

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

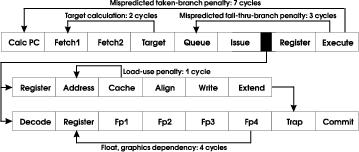


Рис. 19

Это один из самых длинных конвейеров среди всех совре-менных универсальных микропроцессоров архитектуры RISC. Общепризнанное слабое место подобного подхода - проблема за-полнения длинных конвейеров. Так, связанная с неправильным

121

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 резуль-таты вычислений в порядке их запуска на исполнение). Аналогич-ное решение (выровненные конвейеры целочисленной и плаваю-щей арифметики) успешно использовалось в предшествующих по-колениях 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 опера-ции, 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 ис-полнительной 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

* том порядке, в котором они поступили на исполнение.
  + отличие от многих других современных процессоров, ис-пользующих механизм произвольного запуска команд на исполне-ние, то есть не в порядке их расположения в программе, UltraSPARC III действует строго по порядку. По утверждениям разработчиков, это позволяет сократить объем логики управления
* 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 предшест-венниках, 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-

122

полнительных устройств); правда, средневзвешенное число од-новременно запускаемых команд равно четырем.

Команды выбираются из I-кэш первого уровня и помеща-ются в очередь команд (буфер) длиной 20 строк (Instruction Queue), откуда группами направляются в соответствующие испол-нительные устройства. Максимальное число команд в группе - 6. Все команды в группе получают идентификационный код, в соот-ветствии с которым на выходе из конвейера будут сложены их ре-зультаты.

Выше уже упоминалось, что время выполнения большинст-ва целочисленных команд существенно меньше длины исполни-тельной части целочисленного конвейера, и что фиксированная длина конвейера обеспечивает сохранение порядка поступления команд. Однако, это не означает, что результаты становятся дос-тупны только по достижению выхода конвейера. На самом деле полученные данные могут быть использованы другими команда-ми, находящимися на стадии выполнения, уже на следующем так-те после получения результата . Это возможно благодаря наличию рабочего регистрового файла, в котором хранятся все промежу-точные результаты вычислений и из которого, по завершению ис-полнения группы команд, эти результаты переписываются в архи-тектурный регистровый файл. Таким образом, реальная эффектив-ность конвейера заметно повышается.

Устройство предсказания ветвлений является непременным атрибутом любого современного высокопроизводительного мик-ропроцессора, однако, в каждом конкретном случае оно реализо-вано по-своему. Основная задача данного устройства состоит в со-кращении накладных расходов из-за нарушений в работе конвейе-ра 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 ре-шили использовать достаточно простой одноуровневый механизм (в отличие от двухуровневого адаптивного механизма в Alpha 21264). Он представляет собой таблицу на 16 К значений (ее раз-

123

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 К6. Надо отметить, что в UltraSPARC III бу-

фер адресов перехода отсутствует, поэтому при выполнении пра-вильно предсказанного перехода возникает дополнительная за-держка.

Помимо механизма предсказания ветвлений в процессоре используется еще и стек адресов возврата на 8 значений, запоми-нающий 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 предска-занное 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 взаимодейст-вует 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 интер-фейс.

Достаточно очевидно, что такая многошинная архитектура способствует более эффективной работе многопроцессорной сис-темы 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 чип-сета и т.п.). Рассмотрим подробнее каждый из перечисленных ин-терфейсов.

Поскольку архитектура процессора подразумевает едино-личное владение памятью, то есть отсутствие непосредственного доступа к ней со стороны каких-либо других устройств, появляет-ся возможность достижения максимальной пропускной способно-сти данного канала, для чего предусмотрен накристальный кон-

124

троллер памяти. Такое решение имеет следующие положитель-ные 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 памя-ти (SRAM), работающих на частоте 200 МГц, пропускная способ-ность данного канала способна достичь 6.4 ГБ/с. И это не предел.

Во-вторых, по аналогии с накристальным SDRAM-контроллером канала основной памяти, данный интерфейс имеет накристальную таблицу тэгов вторичного кэш. Размер таблицы со-ставляет 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 внекристального расположения таблицы тэгов. Соот-ветственно , при непопадании в кэш процедура инициализации об-ращения к основной памяти начинается на несколько тактов рань-ше. 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 понят-но, зачем это нужно.

125

Сначала сохраняемая информация записывается в оче-редь. Это происходит во время выполнения команды сохранения. Затем, после завершения команды , данные записываются в L1 и, одновременно, в кэш записи. При этом, если происходит непопа-дание в L1, то его содержимое не обновляется. В противном слу-чае из-за сквозного режима обновления данной кэш-памяти про-исходило 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 ор-ганизации канала записи позволяет сократить трафик на шине вто-ричной кэш-памяти на 90%.

Системный интерфейс по своим характеристикам аналоги-чен каналу основной памяти. Из специфических механизмов, свойственных только ему, следует отметить поддержку многопро-цессорности (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 встраивае-мых системах высших моделей, но его основное назначение – вы-сокопроизводительные 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 "производи-

тельные вычисления" (Throughput Computing). Процессор UltraSpark IV совместим на уровне двоичных кодов со Spark-процессорами предыдущих поколений и поддерживает 8 и 9 вер-

сии ОС Solaris v.8 и 9.

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

126

ет четыре команды/такт. Шести исполнительным устройствам параллельного действия (двум целочисленным устройствам, одно-му 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 – пред-варительной выборки. Как и все современные 64-бит микропро-цессоры, UltraSpark IV имеет внутрисхемный контроллер внешней оперативной памяти (статического ДОЗУ). Нити (процессы), вы-полняемые UltraSpark IV, совместно используют адресную шину и шину данных для получения доступа к ячейкам L2 кэш (типа СО-ЗУ) объемом 8 Мбайт, к контроллеру оперативной памяти и со-единительной шине типа Sun Fireplane с максимальной пропуск-ной способностью 2,4 Гбайт/с . Максимальная мощность, потреб-ляемая 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 из-делий). Он содержит 66 млн. транзисторов. Монтируется в 1368-выводной типа LGA.

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

Фирмой планируется создание процессора (кодовое назва-ние Niagara) с восемью процессорными ядрами на кристалле , од-новременно обрабатывающими в целом до 32 нитей. Можно с уве-ренностью сказать, что компания Sun Microsystems хорошо пони-мает, что понадобится в ближайшие годы ей и ее заказчикам.

127

**6.4. Alpha 21264 DEC**

Развитие МП Alpha сначала шло в направлении первооче-редного роста тактовой частоты при относительно простой микро-архитектуре МП. Наиболее ярко это выразилось в Alpha 21064. Микроархитектура 21164 сильно усложнилась, а сам МП далеко обогнал всех оппонентов по тактовой частоте и пиковой произво-дительности при работе с вещественной арифметикой. При этом это 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 под-держивают внеочередное выполнение команд. С другой стороны, нельзя сказать, что компания DEC шла по неверному пути, когда стремилась в первую очередь увеличивать тактовую частоту, со-храняя относительно более простую архитектуру МП Alpha. Дока-зательством 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 пере-ходов.

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 созда-нию уникального интегрированного на чипе двухуровневой кэш, включающей прямоадресуемые I- и D-кэш L1 емкостью 8 Кбайт каждый , плюс 3-канальный частично-ассоциативный кэш L2 емко-стью 96 Кбайт.

128

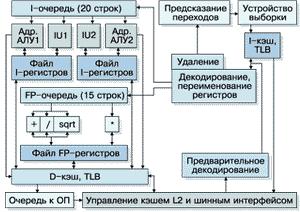


Рис. 20

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 прямоад-ресуемый, 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. Элимини-руется 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 взаимо-действующие блоки микропроцессора, располагающиеся рядом. D-кэш должен работать как с целочисленными ФИУ и ФИУ с пла-вающей запятой , так и с системной шиной. За увеличение емкости первичного кэш в 21264 пришлось заплатить увеличением време-

129

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 ком-пенсируется его большей емкостью и одноуровневой организаци-ей. DEC оценивает уменьшение общей пиковой производительно-сти из-за этого дополнительного такта в 4%. Альтернативные под-ходы привели бы к большему ее снижению.

D-кэш L1 в Alpha 21264 является двухпортовым и позволя-ет 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 блоки-руют 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 Аlpha 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 исполь-зовать 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, че-му отвечает пропускная способность кэша 3.2-4.0 Гбайт/с. Отме-тим , 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 5 нс- SRAM, но за 1 такт через выходные шины отсылают данные

130

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 %. За-держка 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 осуще-ствляется 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 переимено-вание регистров на лету. Если Alpha 21164 имеет 32 целочислен-ных 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. Вне-очередное 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 вклю-чены также команды загрузки регистров/записи в память. Коман-ды выбираются из I-кэша, куда они попадают частично декодиро-ванными. Предварительное декодирование создает 3-разрядное поле, по которому определяется, к какому ФИУ cледует направить команду: целочисленному ФИУ общего назначения, (целочислен-ному) 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 - об-щего назначения, и 2 - адресных АЛУ. Последние отвечают за вы-полнение всех команд загрузки/записи в память и могут также вы-полнять простые арифметические и логические целочисленные

131

операции. 2 ФИУ общего назначения производят арифметиче-ские и логические операции над целыми числами, операции сдвига и перехода. Одно из них умеет умножать, а другое - выполнять но-вые команды из мультимедийного расширения набора команд

MVI (Motion-Video Instructions).

После окончательного декодирования "целочисленные" ко-манды отправляются в очередь емкостью 20 строк. Простые цело-численные операции, например, сложение, могут выполняться в любых целочисленных ФИУ. С целью упрощения логики уже при постановке в очередь решается, направлять ли команду в целочис-ленные ФИУ общего назначения или в адресные АЛУ. Каждый такт команды из очереди, операнды которых доступны, арбитри-руются на предмет направления в ФИУ. Выборка команд осущест-вляется не обязательно в соответствии с их расположением в про-грамме.

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 техно-логическим причинам продублировали файл целочисленных реги-стров. Каждая копия файла имеет 4 порта чтения и 6 портов запи-си. Каждому файлу ставится в соответствие свое целочисленное ФИУ общего назначения и АЛУ. Такой комплект оборудования DEC называла "кластером". Файлы регистров остаются синхрони-зированными, но это требует одного дополнительного такта для записи из АЛУ одного кластера в файл регистров другого класте-ра. 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 c плавающей запятой в Alpha 21264 всего 2. Их кон-вейеры имеют длину 4 такта, что больше, чем у HP PA-8000 (3 такта ) и MIPS R10000 (2 такта). Представление об основных ста-диях 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 ре-гистров/записи в память, 25% умножений и 25% сложений . Как и MIPS R10000, Alpha 21264 может выдавать 2 результата с пла-

132

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 конвейери-зованы и занимают 16 и 33 такта соответственно.



Рис. 21

Важнейшим компонентом современных суперскалярных микропроцессоров является предсказание переходов. Неправиль-ное предсказание заметно ухудшает производительность . Так, средняя задержка при неправильном предсказании в Alpha 21264 составляет свыше 11 тактов. Особенно важным является предска-зание переходов для программ, ориентированных на решение коммерческих задач, для которых характерен большой процент команд условного перехода. Алгоритмы предсказания быстро раз-виваются; одни методы предсказания переходов оказываются точ-нее других, но разные алгоритмы хорошо работают на разных ти-пах переходов.

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

133

перехода динамически выбирается лучший алгоритм. Эта таб-лица 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 досту-па к I-кэшу приходится предпринимать особые меры по уменьше-нию задержек, связанных с переходом. Каждая строка в I-кэше со-держит 4 команды вместе с дополнительными полями, относящи-мися к предсказанию, какая строка будет выбираться следующей. Преимуществом 21264 является то, что выборка команд происхо-дит 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 адре-сов возврата, применяемый для организации эффективного выхода из вызванных подпрограмм. Этот стек больше, чем в других МП, и позволяет аккуратно отслеживать до 32 уровней вызова подпро-грамм.

Минусом микроархитектуры Alpha 21264 является группи-ровка команд в "четверки" и необходимость их соответствующего выравнивания для эффективной обработки переходов. Так, жела-тельно , чтобы адрес перехода указывал на первую команду чет-верки.

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 способно-стью 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 часто-ты процессора.

Фактически эта шина осуществляет соединение "точка-точка", что говорит об ориентации на применение коммутаторов вместо традиционной системной шины. Коммутаторы использу-ются в известных многопроцессорных cистемах: NUMA-системах

SGI Origin 2000, cерверах cерии SPP от Convex/HP и др. Можно предположить, что подобное строение будет применяться и в бу-

134

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 способ-ность самой системы памяти разделяется процессорами. Систем-ная 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 процес-сорами; возможно создание "4-процессорной" версии.

Набор микросхем Цунами позволяет демультиплексировать 64-разрядные данные шины в 128-, 256- и даже 512-разрядный ин-терфейс 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 микро-схем 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 внекристаль-ная реализация кэш, что позволяет реализовать различные объемы кэш-памяти и оптимизировать конструкцию в зависимости от ус-ловий применения (рис. 22). Хранение команд и данных осуществ-ляется в раздельных кэшах, причем процессор соединяется с ними с помощью высокоскоростных 64-битовых шин.

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

135

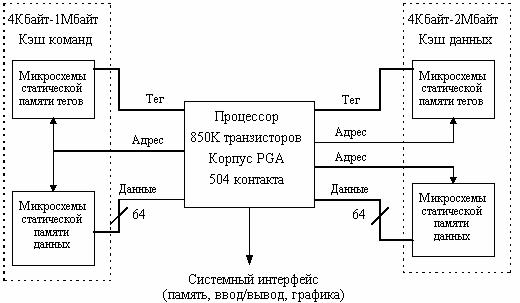


Рис. 22

Микропроцессор аппаратно поддерживает различный объ-ем кэш -памяти: кэш команд может иметь объем от 4 Кбайт до 1 Мбайт, кэш данных - от 4 Кбайт до 2 Мбайт. Чтобы снизить коэф-фициент промахов применяется механизм хеширования адреса. В обоих кэшах для повышения надежности применяются дополни-тельные контрольные разряды, причем ошибки кэша команд кор-ректируются аппаратными средствами.

Процессор подсоединяется к памяти и подсистеме вво-да/вывода посредством синхронной шины. Процессор может рабо-тать с тремя разными отношениями внутренней и внешней такто-вой 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 целочис-ленный 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-

136

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

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

В состав процессора плавающей точки входят устройство умножения, арифметико-логическое устройство, устройство деле-ния 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 арифме-тику с одинарной и двойной точностью в стандарте IEEE 754.

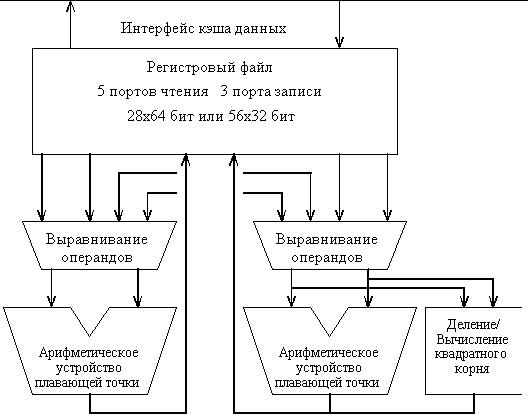


Рис. 23

137

Его устройство умножения используется также для вы-полнения операций целочисленного умножения. Устройства деле-ния и вычисления квадратного корня работают с удвоенной часто-той процессора. Арифметико-логическое устройство выполняет операции сложения, вычитания и преобразования форматов дан-ных. Регистровый файл состоит из 28 64-битовых регистров, каж-дый из которых может использоваться как два 32-битовых регист-ра для выполнения операций с плавающей точкой одинарной точ-ности. Регистровый файл имеет пять портов чтения и три порта записи, которые обеспечивают одновременное выполнение опера-ций умножения, сложения и загрузки/записи.

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

Конвейер целочисленного устройства включает шесть сту-пеней: чтение из кэша команд (IR), чтение операндов (OR), вы-полнение/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 не-большого буфера предварительной выборки , который обеспечива-ет предварительную выборку команд за два такта до начала рабо-ты ступени IR. Во время выполнения на ступени OR все исполни-тельные устройства декодируют поля операндов в команде и на-чинают вычислять результат операции. На ступени DR целочис-ленное устройство завершает свою работу . Кроме того, кэш-память данных выполняет чтение, но данные не поступают до мо-мента 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 коман-дами записи (STORE) требует двух тактов. Наиболее раннее двух-тактное окно команды STORE возникает на ступенях RW и DW. Однако это окно может сдвигаться, поскольку записи в кэш дан-ных происходят только когда появляется следующая команда за-писи. Операции деления и вычисления квадратного корня для чи-

138

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

Конвейер проектировался с целью максимального увеличе-ния времени, необходимого для выполнения чтения внешних кри-сталлов SRAM кэш-памяти данных. Это позволяет максимизиро-вать частоту процессора при заданной скорости SRAM. Все ко-манды загрузки (LOAD) выполняются за один такт и требуют только одного такта полосы пропускания кэш-памяти данных. По-скольку кэши команд и данных размещены на разных шинах, в конвейере отсутствуют какие-либо потери, связанные с конфлик-тами 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 следова-ния пары команд, которые выполняются вместе. Кроме того, от-сутствуют потери тактов, связанных с переключением с выполне-ния 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 до-биться этого, в кэше команд был реализован специально предна-значенный для этого заранее декодируемый бит, чтобы отделить команды целочисленного устройства от команд устройства пла-вающей точки. Этот бит предварительного декодирования команд минимизирует время, необходимое для правильного разделения команд.

Потери, связанные с зависимостями по данным и управле-нию, в этом конвейере минимальны. Команды загрузки выполня-ются 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 производи-тельности циклов передачи управления вперед по программе про-гнозируются как невыполняемые переходы, а передачи управле-

139

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

Количество тактов, необходимое для записи слова или двойного слова командой STORE, уменьшено с трех до двух так-тов. В более ранних реализациях архитектуры PA-RISC был необ-ходим один дополнительный такт для чтения тега кэш, чтобы га-рантировать попадание, а также для того, чтобы объединить ста-рые данные строки кэш- памяти данных с записываемыми данны-ми. PA 7100 использует отдельную шину адресного тега, чтобы совместить по времени чтение тега с записью данных предыдущей команды STORE. Кроме того, наличие отдельных сигналов разре-шения записи для каждого слова строки кэш-памяти устраняет не-обходимость объединения старых данных с новыми, поступаю-щими при выполнении команд записи слова или двойного слова.

Этот алгоритм требует, чтобы запись в микросхемы SRAM происходила только после того, когда будет определено, что дан-ная запись сопровождается попаданием в кэш и не вызывает пре-рывания . Это требует дополнительной ступени конвейера между чтением тега и записью данных. Такая конвейеризация не приво-дит к дополнительным потерям тактов, поскольку в процессоре реализованы специальные цепи обхода, позволяющие направить отложенные данные команды записи последующим командам за-грузки или командам STORE, записывающим только часть слова. Для данного процессора потери конвейера для команд записи сло-ва или двойного слова сведены к нулю, если непосредственно по-следующая 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 большинст-во команд записи в действительности работают с однословным или двухсловным форматом.

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

140

каждом такте при отсутствии каких-либо потерь. Последова-тельные операции с зависимостями по регистрам приводят к поте-ре одного такта. Команды деления и вычисления квадратного кор-ня 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 следую-щая команда деления/вычисления квадратного корня.

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

Потери для операций плавающей точки, использующих предварительную загрузку операнда командой LOAD, составляют один такт, если команды загрузки и плавающей арифметики явля-ются 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, мини-мизируются посредством применения четырех разных методов: "попадание при промахе" для команд LOAD и STORE, потоковый режим работы с кэшем данных, специальная кодировка команд за-писи, позволяющая избежать копирования строки, в которой про-изошел промах, и семафорные операции в кэш-памяти. Первое свойство позволяет во время обработки промаха в кэш данных вы-полнять 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.

141

Компилятор может использовать это свойство для пред-варительной выборки в кэш необходимых данных задолго до того момента, когда они действительно потребуются. Для промахов, возникающих при выполнении команды STORE, обработка после-дующих команд загрузки или операций записи в части одного сло-ва продолжается до тех пор, пока не возникает обращений к стро-ке, в которой произошел промах. Компилятор может использовать это свойство для выполнения команд на фоне записи результатов предыдущих вычислений. Во время задержки, связанной с обра-боткой промаха, другие команды LOAD и STORE, для которых происходит попадание в кэш данных, могут выполняться и другие команды целочисленной арифметики и плавающей точки. В тече-ние 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, пре-дотвращающий 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" по-зволяет продолжить выполнение, как только нужное слово или двойное слово возвращается из памяти . Таким образом, выполне-ние команд может продолжаться как во время задержки, связанной с обработкой промаха, так и во время заполнения соответствую-щей строки при промахе.

При выполнении блочного копирования данных в ряде случаев компилятор заранее знает, что запись должна осуществ-ляться 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

142

времени , необходимого для записи нового тега кэш. В процес-соре PA 7100 такая возможность реализована как для привилеги-рованных, так и для непривилегированных команд.

Последнее улучшение управления кэш-памятью данных связано с реализацией семафорных операций "загрузки с обнуле-нием" непосредственно в кэш -памяти. Если семафорная операция выполняется в кэш, то потери времени при ее выполнении не пре-вышают потерь обычных операций записи. Это не только сокра-щает конвейерные потери, но и снижает трафик шины памяти. В архитектуре предусмотрен также другой тип специального коди-рования 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 от-сутствующей в кэше команды из памяти. 64- битовая магистраль данных, используемая для заполнения блоков кэша команд, соот-ветствует максимальной полосе пропускания внешней шины па-мяти 400 Мбайт/с при тактовой частоте 100 МГц.

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

Конструкция процессора обеспечивает реализацию двух способов построения многопроцессорных систем. При первом способе каждый процессор подсоединяется к интерфейсному кри-сталлу, который наблюдает за всеми транзакциями на шине основ-ной памяти. В такой системе все функции по поддержанию коге-рентного состояния кэш-памяти возложены на интерфейсный кри-сталл, который посылает процессору соответствующие транзак-ции. Кэш данных построен на принципах отложенного обратного копирования и для каждого блока кэш-памяти поддерживаются биты состояния "частный" (private), "грязный" (dirty) и "достовер-ный" (valid), значения которых меняются в соответствии с тран-закциями, которые выдает или принимает процессор.

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

143

сталлов и она совместима с существующей системой памяти. Когерентность кэш-памяти обеспечивается наблюдением за ло-кальной шиной памяти. Пересылки строк между кэшами выпол-няются без участия контроллера памяти и ввода-вывода. Такая конфигурация обеспечивает возможность построения очень деше-вых высокопроизводительных многопроцессорных систем.

Процессор поддерживает ряд операций, необходимых для улучшения графической производительности рабочих станций се-рии 700: блочные пересылки, Z-буфери-зацию, интерполяцию цве-тов 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 так-товую частоту 100 МГц.

**6.6. R12000 MIPS**

**Организация памяти микропроцессора R12000.** MIPSR12000 использует два уровня кэш-памяти. Первый уровень на-борно 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 мо-жет 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 па-раллельная 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 прин-ципу "запрошенное слово первым", что позволяет существенно со-кратить простои процессора из-за ожидания требуемой информа-ции. 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

144

банков емкостью по 16 Кбайт, что обеспечивает двукратное расслоение при выполнении обращений к этой кэш-памяти. Каж-дый 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 ко-тором 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 незави-симыми . Эти четыре массива работают под общим управлением очереди формирования адресов памяти и схем внешнего интер-фейса кристалла. В очереди адресов могут одновременно нахо-диться до 16 команд загрузки и записи, которые обрабатываются в четырех отдельных конвейерах. Команды из этой очереди динами-чески подаются для выполнения в специальный конвейер, который обеспечивает вычисление исполнительного виртуального адреса и преобразование этого адреса в физический. Три других параллель-но работающих конвейера могут одновременно выполнять про-верку тегов, осуществлять пересылку данных для команд загрузки и завершать выполнение команд записи в память. Хотя команды выполняются в строгом порядке их расположения в памяти, вы-числение адресов и пересылка данных для команд загрузки могут происходить неупорядоченно. Схемы внешнего интерфейса кри-сталла могут производить заполнение или обратное копирование строк кэш-памяти, либо операции просмотра тегов. Такая парал-лельная работа большинства устройств процессора позволяет про-цессору R12000 эффективно выполнять реальные многопроцес-сорные приложения.

Работа конвейеров кэш-памяти данных тесно координиро-вана. Например, команды загрузки могут выполнять проверку те-гов и чтение данных в том же такте, что и преобразование адреса. Команды записи сразу же начинают проверку тегов, чтобы, в слу-чае необходимости, как можно раньше инициировать заполнение требуемой строки изL2, но непосредственная запись данных в кэш задерживается до тех пор, пока сама команда записи не станет са-

145

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 од-новременно с заполнением строки кэш -памяти данные могут по-ступать по цепям обхода в регистровый файл.

При обнаружении промаха при обращении к кэш-памяти данных ее работа не блокируется, т.е. она может продолжать об-служивание 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 не-больших 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 процес-сора. Не требуется никаких внешних интерфейсных схем. Мини-мальный объем L2 составляет 512 Кбайт, максимальный размер - 16 Мбайт. Размер строки этой кэш-памяти программируется и мо-жет составлять 128 или 1024 байт.

Одним из методов улучшения временных показателей рабо-ты 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 соответст-вует 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 процессор-ном кристалле. Поэтому после обнаружения промаха в первичном

146

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 ин-вертированием 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 боль-шой емкости обычной практикой является использование кодов, исправляющих одиночные ошибки (ЕСС-кодов). В R12000 с каж-дой четырехсловной строкой хранится 9-разрядный ЕСС-код и бит четности. Дополнительный бит четности позволяет сократить за-держку (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 процес-се ее загрузки команды частично декодируются. При этом к каж-дой команде добавляются 4 дополнительных бита, указывающих на исполнительное устройство, в котором будет выполняться ко-манда. Таким образом, в кэш-памяти команды хранятся в 36-разрядном формате.

**6.7. PowerPC 970 IBM**

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

147

SIMD-поддержкой). Это совсем неплохо для архитектуры, со-вместимой с 32-бит процессорами семейства. Компания IBM по-шла 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 конст-рукции процессора, на одном кристалле можно разместить два процессорных ядра с более чем десяток исполнительных уст-ройств. Архитектура процессоров PowerPC оперирует более мощ-ными и гибкими SIMD-командами, чем другие устройства, пред-назначенные для мультимедийной обработки. Altivec SIMD-поддержка обеспечивает еще большую универсальность, предос-тавляя 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 микро-процессора Power4 и добавление векторной Altivec-поддержки по-зволили 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 такто-вую 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 85х85 мм. Четыре таких модуля могут быть объ-единены без связующих логических схем. Получаемая в результа-те 32-канальная микропроцессорная субсистема имеет более 20 тыс. контактных площадок ввода-вывода и рассеивает ~2 кВт.

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

148

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 ке-рамический BGA-корпус размером 25х25 мм.

В результате разработчики получили высокопроизводи-тельный , 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 много-кристальный модуль. Наряду с этим, у него более глубокий кон-вейер (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 про-цессора PowerPC 970FX по 90-нм КНИ-технологии в сочетании с технологиями напряженного кремния и восьмислойных медных соединений. В микросхеме использован новый метод сложной на-стройки и управления тактовой частотой и пороговым напряжени-ем транзисторов, позволяющий регулировать эти параметры с ша-гом 0,5 МГц и 0,5 мВ, соответственно. В результате можно полу-чать четыре различных значения порогового напряжения транзи-сторов с тонким затворным окислом и еще два значения напряже-ния 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 связных сис-тем. 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.

149

**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 об-работку данных. Данные должны быть представлены в виде огра-ниченного количества форматов (например , массивы, списки, за-писи), должна быть явно создана структура связей между элемен-тами 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 сис-тем программирования, а также служит препятствием к созданию вычислительных средств с архитектурой, ориентированной на бо-лее эффективное использование параллелизма обработки данных.

Ассоциативный способ обработки данных позволяет пре-одолеть многие ограничения, присущие адресному доступу к па-мяти, 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 выделе-ния искомых данных из всех данных. Поиск данных может проис-ходить по фрагменту, имеющему большую или меньшую корреля-цию с заданным элементом данных.

Исследованы и в разной степени используются несколько подходов, различающихся полнотой реализации модели ассоциа-тивной обработки. Если реализуется только ассоциативная выбор-ка данных с последующим поочередным использованием найден-ных данных, то говорят об ассоциативной памяти или памяти, ад-ресуемой по содержимому. При достаточно полной реализации всех свойств ассоциативной обработки, используется термин «ас-социативный процессор».

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

150

ных устройств, способных одновременно по командам управ-ляющего устройства вести обработку нескольких потоков данных. В ассоциативных вычислительных системах информация на обра-ботку 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 матрич-ные 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 эле-ментов, 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 эффектив-ность системы при решении широкого круга задач необходимо ор-ганизовать 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 SОLОМОN (60-е годы) (рис. 24). Система SОLOМОN содержит 1024 процес-сорных элемента, соединены в виде матрицы: 32х32. Каждый про-цессорный элемент матрицы включает в себя процессор, обеспе-чивающий выполнение последовательных поразрядных арифмети-ческих и логических операций, а также оперативное ЗУ, емкостью 16 Кбайт. Длина слова - переменная от 1 до 128 разрядов. Разряд-ность слов устанавливается программно. По каналам связи от уст-ройства управления передаются команды и общие константы.

151

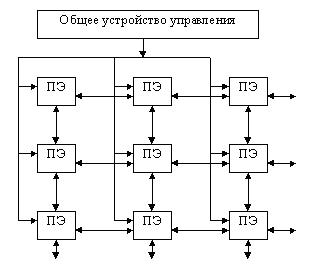


Рис. 24

В процессорном элементе используется, так называемая, многомодальная логика, которая позволяет каждому процессорно-му элементу выполнять или не выполнять общую операцию в за-висимости от значений обрабатываемых данных. В каждый мо-мент 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 состоя-ния - 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 выпол-няет 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. Взаимодейству-ют процессорные элементы с периферийным оборудованием через внешний процессор.

152

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

* 1. **ДНК процессоры**
* настоящее время в поисках реальной альтернативы полу-проводниковым технологиям создания новых вычислительных систем ученые обращают все большее внимание на биотехноло-гии, или биокомпьютинг, который представляет собой гибрид ин-формационных, молекулярных технологий, также биохимии. Био-компьютинг позволяет решать сложные вычислительные задачи, пользуясь методами, принятыми в биохимии и молекулярной био-логии, организуя вычисления при помощи живых тканей, клеток, вирусов и биомолекул.

Наибольшее распространение получил подход, где в каче-стве основного элемента (процессора) используются молекулы де-зоксирибонуклеиновой кислоты. Центральное место в этом подхо-де занимает так называемый ДНК - процессор. Кроме ДНК в каче-стве био-процессора могут быть использованы также белковые

молекулы и биологические мембраны. Так же, как и любой другой процес-



сор, ДНК процессор характеризуется структурой и набором команд. В нашем случае структура процессора - это структу-ра молекулы ДНК. А набор команд - это перечень биохимических операций с моле-

кулами.

153

Принцип устройства компьютерной ДНК-памяти осно-ван на последовательном соединении четырех нуклеотидов (ос-новных кирпичиков ДНК-цепи). Три нуклеотида, соединяясь в любой последовательности, образуют элементарную ячейку памя-ти - кодон, которые затем формируют цепь ДНК. Основная труд-ность в разработке ДНК-компьютеров связана с проведением из-бирательных однокодонных реакций (взаимодействий) внутри це-пи 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 яв-ляется самосборка ДНК, приводящая к потере информации. Ее преодолевают введением в клетку специальных ингибиторов - ве-ществ, предотвращающих химическую реакцию самосшивки.

Использование молекул DNA для организации вычислений

– это не слишком новая идея. Теоретическое обоснование подоб-ной 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 про-бирке молекулу ДНК, в которой были закодированы исходные данные, и специальным образом подобранные ферменты. В ре-зультате 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, исполь-зуя ДНК, кодировать двоичные числа и решать проблему удовле-творения логического выражения. Суть ее в том, что, имея некото-рое логическое выражение, включающее *n* логических перемен-ных, нужно найти все комбинации значений переменных , делаю-щих выражение истинным. Задачу можно решить только перебо-ром *2n* комбинаций. Все эти комбинации легко закодировать с по-мощью ДНК, а дальше действовать по методике Адлемана.

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

154

института естественных наук. Она имитировала работу “моле-кулярной машины” в живой клетке, собирающей белковые моле-кулы по информации с ДНК, используя РНК в качестве посредни-ка 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 зависимо-сти 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, зако-дированной в цепочках химических соединений.

Разработанная в Вейцмановском институте установка коди-рует входные данные и программы в состоящих из двух цепей мо-лекулах ДНК и смешивает их с двумя ферментами. Молекулы фермента выполняли роль аппаратного, а молекулы ДНК - про-граммного 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 конечно-му 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 крем-ниевых 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 выпол-няться 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-

155

ществляет химические реакции между молекулами ДНК, обес - печивает поиск и выделение результата вычислений. Вторая - об-рабатывает информацию и анализирует полученные результаты.

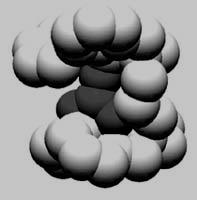
Возможностями биокомпьютеров заинтересовались и воен-ные. Американское агентство по исследованиям в области оборо-ны DARPA выполняет проект, получивший название Bio-Comp (Biological Computations, биологические вычисления). Его цель - создание мощных вычислительных систем на основе ДНК.

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 неду-гов.

**7.4. Клеточные процессоры**

Клеточные процессоры представляют собойсамооргани-зующиеся колонии различных "умных" микроорганизмов, в геном которых удалось включить некую логическую схему, которая мог-ла бы активизироваться в присутствии определенного вещества. Для этой цели идеально подошли бы бактерии, стакан с которыми

и представлял бы собой компью-тер. Такие компьютеры очень де-шевы в производстве. Им не нужна столь стерильная атмосфера, как при производстве полупроводни-ков.



Главным свойством процес-сора такого рода является то, что каждая их клетка представляет со-бой миниатюрную химическую ла-бораторию. Если биоорганизм за-

программирован, то он просто производит нужные вещества. Дос-таточно вырастить одну клетку, обладающую заданными качест-вами, и можно легко и быстро вырастить тысячи клеток с такой же программой.

156

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

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

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

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

1. более простая технология изготовления, не требующая для своей реализации столь жестких условий, как при производст-ве полупроводников
2. использование не бинарного, а тернарного кода (инфор-мация кодируется тройками нуклеотидов), что позволит при меньшем количестве шагов перебрать большее число вариантов при анализе сложных систем

157

1. потенциально исключительно высокая производи-тельность, которая может составлять до 1014 операций в секунду за счет одновременного вступления в реакцию триллионов моле-кул ДНК
2. возможность хранить данные с плотностью, в триллионы раз превышающей показатели оптических дисков
3. исключительно низкое энергопотребление

Однако , наряду с очевидными достоинствами, биокомпью-теры имеют и существенные недостатки, такие как:

1. сложность со считыванием результатов - современные способы определения кодирующей последовательности не совер-шенны, сложны, трудоемки и дороги
2. низкая точность вычислений, связанная с возникновени-ем мутаций, прилипанием молекул к стенкам сосудов и т.д.
3. невозможность длительного хранения результатов вы-числений в связи с распадом ДНК в течение времени

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

**7.5. Коммуникационные процессоры**

Коммуникационные процессоры - это микрочипы, являю-щие 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, опти-мизированы для сетевой работы, и на их основе производители - как процессоров, так и оборудования - пишут программное обес-печение для специфических приложений.

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

158

ми узлами. Скоростной коммуникационный процессор с RISC-ядром позволяет управлять обменом данными по нескольким не-зависимым каналам, поддерживать практически все распростра-ненные протоколы обмена, гибко и эффективно распределять и обрабатывать последовательные потоки данных с временным раз-делением каналов.

Сама идея создания процессоров, предназначенных для оп-тимизации сетевой работы - и при этом достаточно универсальных для программной модификации – родилась в связи с необходимо-стью 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 се-тевых приложениях, но и для построения интернет-ориентированных встраиваемых систем промышленного назначе-ния.

Эффективность работы промышленных предприятий сего-дня напрямую зависит от гибкости применяемых систем автомати-зированного 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 удален-ных диспетчерских пунктов. Осуществление этих целей возможно с помощью коммуникационных процессоров, предназначенных для подключения персональных компьютеров к промышленным информационным сетям.

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

159

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, предна-значенные для выполнения всех или некоторых функций систем управления базами данных (СУБД). Если в свое время системы управления базами данных предназначались в основном для хра-нения текстовой и числовой информации, то теперь они рассчита-ны на самые различные форматы данных, в том числе графиче-ские, звуковые и видео. Процессоры баз данных выполняют функ-ции управления и распространения, обеспечивают дистанционный доступ к информации через шлюзы, а также репликацию обнов-ленных данных с помощью различных механизмов тиражирова-ния.

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 пользо-вателям возможность в любой момент обратиться к корпоратив-ным данным и проанализировать их, вне зависимости от того, где эти данные размещаются.

Решение таких задач требует существенного увеличения производительности таких систем. Однако традиционная про-граммная реализация многочисленных функций современных СУБД на ЭВМ общего назначения приводит к громоздким и не-производительным системам с недостаточно высокой надежно-стью. Необходим поиск новых архитектурных и аппаратных ре-шений. Интенсивные исследования, проводимые в этой области в настоящее время , привели к пониманию необходимости использо-вания в качестве процессоров баз данных специализированных па-раллельных вычислительных систем. Создание такого рода систем связывается с реализацией параллелизма при выполнении после-довательности операций и транзакций, а также конвейерной пото-ковой обработки данных.

160

**7.7. Потоковые процессоры**

Потоковыми называют процессора, в основе работы кото-рых 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, фи-нансовых расчетов и научных задач. Кроме того, усовершенство-ваны возможности 64-разрядной технологии MMX (целочислен-ных SIMD-команд); эта технология распространена на 128-разрядные числа, что позволяет ускорить обработку видео, речи, шифрование, обработку изображений и фотографий. Потоковый процессор повышает общую производительность, что особенно важно при работе с 3D-графическими объектами.

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

Ярким представителем потоковых процессоров является семейство процессоров Intel, начиная с Pentium III, в основе рабо-

ты которых лежит технология Streaming SIMD Extensions (SSE,

потоковая обработка по принципу "одна команда - много дан-ных"). Эта технология позволяет выполнять такие сложные и не-обходимые в век Internet задачи, как обработка речи, кодирование и декодирование видео- и аудиоданных, разработка трехмерной графики и обработка изображений.

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

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

161

данных. Векторная обработка увеличивает производительность процессора за счет того, что обработка целого набора данных ( век-тора) производится одной командой. Векторные компьютеры ма-нипулируют массивами сходных данных подобно тому, как ска-лярные машины обрабатывают отдельные элементы таких масси-вов. 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 тео-рия 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, суще-ственно расширяют область применения нечеткой логики.

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

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

162

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

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

Задачи с помощью нечёткой логики решаются по следую-щему принципу:

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

Оказалось возможным создание нечеткого процессора, по-зволяющего выполнять различные нечеткие операции и прибли-женные рассуждения (нечеткий вывод) в соответствии с правила-ми логического вывода. В 1986 году в AT&T Bell Labs создавались процессоры с “прошитой” нечеткой логикой обработки информа-ции.

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

163

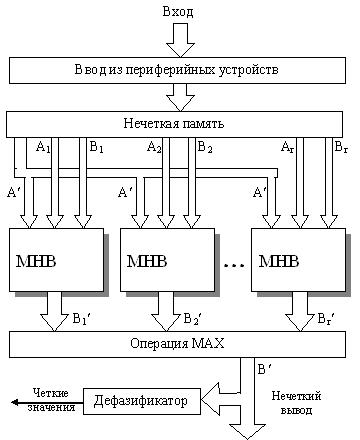


Рис. 25

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

164

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.

* Европе и США ведутся интенсивные работы по интегра-ции fuzzy команд в ассемблеры промышленных контроллеров встроенных устройств (чипы Motorola 68HC11. 12. 21). Такие ап-паратные средства позволяют в несколько раз увеличить скорость выполнения приложений и компактность кода по сравнению с реализацией на обычном ядре. Кроме того, разрабатываются раз-личные варианты fuzzy- сопроцессоров, которые контактируют с центральный процессор через общую шину данных, концентриру-ют свои усилия на размывании/ уплотнении информации и опти-мизации 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 ре-шения (fuzzy-вычислители) позволят получить реальные преиму-щества 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».

* + настоящее время перспективой использовать процессо-ры, основанные на нечеткой логике всерьез заитересовались воен-ные. Известно, что NASA рассматривает возможность применения (если еще не применяет) нечеткие системы для управления про-цессами стыковки космических аппаратов.
    1. **Сигнальные процессоры**
* ответ на возросшие запросы потребителей фирма Motorola разработала новую архитектуру микросхемы, ориентиро-ванную как на выполнение сложных алгоритмов цифровой обра-ботки сигналов, так и на решение задач управления. Семейство микросхем DSP568xx построено на базе ядра 16-разрядного про-

165

цессора DSP56800 с фиксированной точкой. Это ядро предна-значено для эффективного решения задач управления и цифровой обработки сигналов. Реализованный в нем набор команд обеспе-чивает цифровую обработку сигналов с эффективностью лучших DSP общего назначения и отвечает требованиям простоты созда-ния компактных программ управления.

Ядро DSP56800 является программируемым 16-разрядным КМОП процессором , предназначенным для выполнения цифровой обработки сигналов в реальном масштабе времени и решения вы-числительных 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 устройст-вах выполняются параллельно. Каждое из устройств имеет свой набор регистров и логику управления и организовано таким обра-зом, что может функционировать независимо и одновременно с тремя другими. Внутренние шины адресов и данных связывают между собой память, функциональные и периферийные устройст-ва (регистры периферийных устройств расположены в области па-мяти). Таким образом, ядро реализует одновременное выполнение нескольких действий: устройство управления выбирает первую команду, устройство генерации адресов формирует до двух адре-сов второй команды, а АЛУ выполняет умножение третьей коман-ды. Есть альтернативная возможность: в третьей команде опера-цию 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 удоб-ства программирования. Ниже приведены некоторые характери-стики сигнального процессора:

• производительность 40 MIPS при тактовой частоте 80 МГц и напряжении питания 2.7…3.6 В,

166

• наличие набора команд совмещенной обработки, имеющих режимы адресации, характерные для программ цифро-вой обработки сигналов,

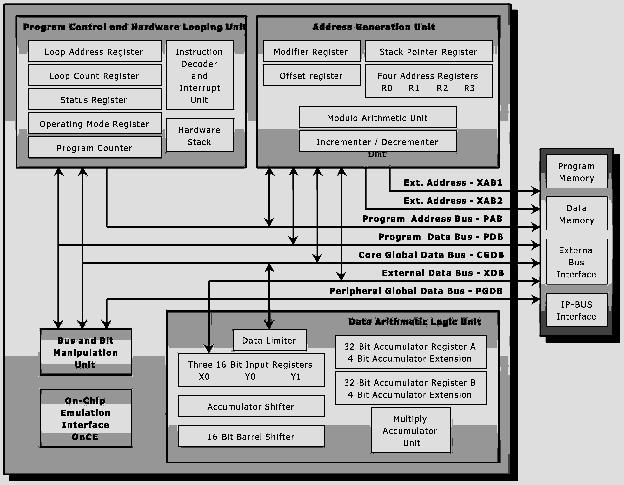


Рис . 26

• однотактный параллельный 16х16 умножитель-сумматор,

167

• два 36-разрядных аккумулятора, включая биты расши-

рения,

• однотактное 16-разрядное устройство циклического сдви-

га,

* 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 процессо-рах, и команды, снижающие объем программы,
* эффективный компилятор языка С и поддержка локаль-ных переменных,
* стек подпрограмм и прерываний, не имеющий ограниче-ния по глубине.

Для любого высокопроизводительного вычислителя, на-пример 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 со-вмещенной с выполнением вычислительных операций передачи данных. Реализованы два типа операций совмещенной передачи - одинарная совмещенная передача и сдвоенное совмещенное чте-ние. Оба типа операций существенно повышают скорость цифро-вой обработки сигналов и численных расчетов. Все команды DSP56800 с совмещенной передачей выполняются за один ко-мандный цикл и занимают одно слово в памяти программ.

Однократная совмещенная передача позволяет выполнить арифметическое действие и одну передачу данных (чтение или за-пись) за один командный цикл. Например, можно одной командой выполнить сложение двух чисел и одновременно данные из реги-стра 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 Х-памяти данных в одной команде за один командный цикл. На-пример, можно в одной команде выполнить умножение двух чи-

168

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

Обычные микроконтроллеры, как правило, имеют объем встроенной в микросхему памяти, достаточный для выполнения сложных алгоритмов управления без использования дополнитель-ной внешней памяти. Многие микросхемы DSP содержат встроен-ную память небольшого объема и, как правило, им требуется внешняя память для хранения программы. Микросхемы же семей-ства DSP56F8хх имеют встроенную память большого объема. Гар-вардская архитектура DSP обеспечивает наличие двух независи-мых областей памяти - данных и программ. Для хранения в микро-схеме данных и программ используется встроенная оперативная память и флэш-память.

Объём памяти каждого типа для микросхем семейства DSP56F8хх приведен в табл. 5. Как память программ, так и память данных могут быть расширены путем подключения внешней па-

мяти. Микросхемы DSP56F803, DSP56F805, DSP56F807 допуска-

ют 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 за- | 2k x 16 | 2k x 16 | 2k x 16 | 2k x 16 |
| грузки |  |  |  |  |

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 со-держали полного набора встроенных периферийных устройств, необходимых для решения задач управления. Использование внешних периферийных устройств приводит к увеличению числа

169

микросхем , усложнению платы и существенному возрастанию стоимости изделия.

Микросхемы семейств DSP56F8хх (рис. 27, табл. 6) имеют широкий набор встроенных периферийных устройств, пригодных для использования в системах управления всех типов

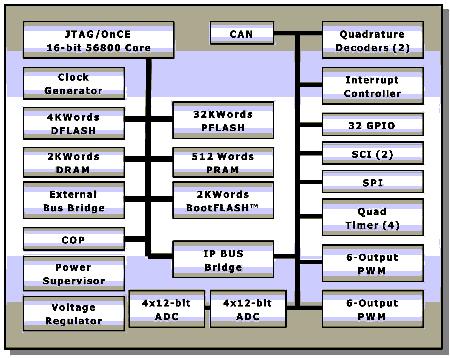


Рис . 27

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 170 |  |
|  |  |  |  | Таблица 6 | |  |
|  | DSP56824 | DSP56F801 | DSP56F803 | DSP56F805 | DSP56F80 |  |
| Тактовая часто- | 70 | 80 | 80 | 80 | 80 |  |
| та, МГц |  |
|  |  |  |  |  |  |
| Производитель- | 35 | 40 | 40 | 40 | 40 |  |
| ность, MIPS |  |
|  |  |  |  |  |  |
| Интерфейсы | 2SPI, SCI, | SCI, SPI | 2SCI, SPI, | 2SCI, SPI, | 2SCI, SPI, |  |
| SSI | CAN | CAN | CAN |  |
|  |  |  |
| АЦП | - | 2 четырех- | 2 четырех- | 2 четырех- | 4 четырех- |  |
| кан. | кан. | кан. | кан. |  |
|  |  | 12-разрядн. | 12-разрядн. | 12-разрядн. | 12-разрядн. |  |
| ШИМ- |  | 6-канальн. | 6-канальн. | 2 6-канальн. | 2 6- |  |
| - | канальн. |  |
| генераторы |  | 15-разр. | 15-разр. | 15-разр. | 15-разр. |  |
|  |  |  |  |  |  |
| Прочие функ- | ФАПЧ, 3- | НВИ\*, | НВИ\*, | НВИ\*, | НВИ\*, |  |
| циональные | кан. тай- | ФАПЧ, 2- | ФАПЧ, 5- | ФАПЧ, 6- | ФАПЧ, 6- |  |
| особенности | мер | кан.таймер | кан.таймер | кан.таймер | 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 уст-ройства имеют заранее определенный интерфейс с ядром DSP (в отличие от внешних периферийных устройств), то упрощаются разработка системы, программирование и управление периферий-ными устройствами. Таким образом, время разработки программ сокращается. Микросхема DSP56F805 содержит следующие пери-ферийные блоки:

* + два шестиканальных ШИМ-генератора (PWMA & PWMB)
* привязкой импульсов к центру или краю временного интервала, программированием длительности "мертвого времени" и защитой в случае возникновения аварийных режимов работы; каждый ге-нератор снабжен тремя сенсорами тока и четырьмя входами ава-рийного отключения,
  + два 12-разрядных АЦП с одновременной выборкой, снаб-женные входными четырехканальными мультиплексорами,

171

* 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 ше-стью входами: таймер C с двумя входами и таймер D с четырьмя входами,
* контроллер CAN интерфейса A/B с двухвыводными пор-тами приемопередатчиков,
* два двухпроводных последовательных коммуникацион-ных интерфейса (SCI0 & SCI1) или 4 дополнительных линии

GPIO,

* последовательный интерфейс периферии (SPI) с настраи-ваемым четырехпроводным портом или четыре дополнительные линии GPIO,
* сторожевой таймер контроля функционирования процес-

сора,

* два программируемых входа внешних прерываний,
* четырнадцать программируемых и восемнадцать мульти-плексированных 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 исполь-зуются прерывания от внешних устройств и внутренних перифе-рийных модулей. Обычно микроконтроллеры поддерживают не-сколько типов внутренних и внешних прерываний и обеспечивают много вариантов маскирования и установки приоритетов.

Обычные DSP обрабатывают только небольшой набор пре-рываний, которые напрямую взаимодействуют с его ядром. В от-личие от них, кристалл DSP56F80х поддерживает большое число прерываний. Хотя число адресуемых прерываний ядра DSP56F8хх мало в сравнении с общим число источников прерываний, много-уровневая встроенная схема мультиплексирования обеспечивает полную и гибкую поддержку 64 источников прерываний, каждый

172

из которых может маскироваться и имеет программно устанав-ливаемый 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 различ-ным быстродействием программируемые временные задержки для памяти программ и памяти данных могут устанавливаться раз-дельно.

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 предшест-вующими архитектурами, не доставят ему беспокойства. Про-граммный 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 ариф-метическими операциями. Эффективная работа трансляторов для микросхем с архитектурой DSP56800 обеспечивается использова-нием в микросхемах команд общего назначения.

Порт отладки JTAG позволяет отлаживать микросхему в составе законченной системы пользователя. Через порт можно за-дать точки останова программы, проверить и изменить содержи-мое регистров и ячеек памяти, выполнить другие действия по от-ладке системы.

Motorola предлагает полный набор программных и аппарат-ных средств быстрой разработки и отладки систем, реализованных на кристаллах семейства DSP568хх. Средства разработки включа-ют:

* оценочные платы для каждой модификации микросхемы,
* интегрированную среду отладки "Metrowerks Code Warrior" со встроенным кросc-компилятором языка С.

Программная среда разработки предоставляет программи-сту гибкое модульное окружение, обеспечивая полное использова-

173

ние возможностей микросхем . Среда допускает различные кон-фигурации памяти данных и позволяет создавать перемещаемый код, выполнять символьную отладку, гибко компоновать объект-ные файлы. Реализованы средства создания архива библиотек при-кладных программ.

Motorola разработала новый комплект для разработки встроенного программного обеспечения (Embedded Software Development Kit, SDK), дополняющий существующую среду раз-работки для DSP568xx. Он формирует программную инфраструк-туру, обеспечивающую разработку высокоэффективных программ, полностью переносимых и допускающих повторное использова-ние не только в процессорах семейства DSP568хх, но в будущем и в процессорах с другой архитектурой, поддерживаемой SDK. Этот программный продукт, выпускаемый для цифровых сигнальных процессоров фирмы Motorola, предназначен для ускорения разра-ботки 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 DSP568хх, имеющими производительность DSP и снабженными набором периферийных устройств, которые тради-ционно используют разработчики систем управления.

Архитектура ядра DSP568хх обеспечивает эффективную цифровую обработку данных и решение задач управления. Такие характеристики этой архитектуры, как высокая производитель-ность и набор команд общего назначения, обеспечивают ей лиди-рующие 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 DSP568хх предназначены для при-менения в недорогих устройствах. Эти микросхемы ориентирова-ны на применение в бытовой технике, для которой необходима низкая стоимость и не требуются высокие параметры. К таким из-делиям относятся:

174

* специализированные и многоцелевые контроллеры,
* проводные и беспроводные модемы,
* системы беспроводной передачи цифровых сообщений,
* цифровые телефонные автоответчики,
* устройства управления серводвигателями и электродвига-телями 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, пото-му что в этом случае уменьшается число внешних компонентов.

* + - 1. **Архитектуры микропроцессорных систем**
  1. **Системы с централизованным, децентрализованным** 
     + - * **комбинированным управлением**
     + МПС с централизованным управлением задача обработки входных сигналов Х1 ,..., Хn с целью формирования управляющих воздействий Y1 ,...,Ym 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.

1. Если осуществляется управление одним, но сложным много-мерным 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 одно-мерных объектов, то система является несвязанной.

175

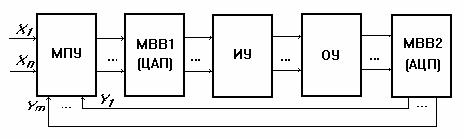


Рис. 28.

* 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 микрокон-троллеры.
* децентрализованных системах МПУ могут вводиться для передачи ему функций диспетчера либо отсутствовать совсем. В этом случае реализуется комбинированное управление. В комби-нированных системах используется обе перечисленные структуры управления.

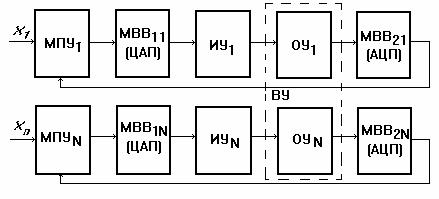


Рис. 29.

Выбор структуры управления в МПС, построенных на базе МПК БИС, зависит от многих взаимосвязанных факторов, важ-нейшими из которых являются стоимость и надежность систем,

176

их живучесть, гибкость, способность работать в масштабе ре-ального времени.

Специфика конкретных задач управления показывает, что применение принципа децентрализованного (распределенного) управления при построении МПС в техническом и экономиче-ском плане имеют преимущества по сравнению с другими струк-турами МПС.

**8.2. Системы с перестраиваемой структурой**

Задачи, решаемые МПС, могут зависеть от характера вход-ных воздействий, поступающих в систему. Так, например, управ-ление роботом может осуществляться по разным алгоритмам в за-висимости от результата решения задачи распознавания представ-ленного роботу объекта. В этом и подобных случаях структура МПС оказывается переменной. В микропроцессорных системах она перестраивается программно.

Пример структурной схемы перестраиваемой МПС управ-ления роботом в общем виде показана на рис. 30.

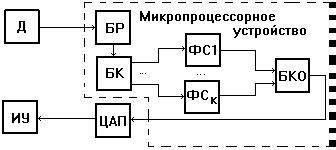


Рис. 30

В состав системы входят: цифровой датчик визуальной ин-формации Д, информация с которого поступает в блок распознава-ния изображения БР, формирующий электрический логотип изо-бражения. Блок коммутации алгоритмов управления БК произво-дит выбор и обработку программы обслуживания сформирован-ного логотипа. С помощью сигналов формирователей управляю-

177

щих сигналов ФС1 , ..., ФСк блок коммутации выхода БКВ вы-дает на выход МПУ исполнительный сигнал, который через ЦАП подается на исполнительное устройство ИУ.

Осуществимость перестройки МПС, выполняемой в реаль-ном масштабе времени на программном уровне, является следст-вием применения в автоматических системах высокопроизводи-тельных МПС, на которые возлагаются задачи обработки больших потоков информации, связанной со статистическим экспресс - анализом случайных сигналов, их идентификацией, классифика-цией, 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 явля-ется резервирование. Резервирование подразделяется на аппара-турное, программное и информационное.

Табл. 7

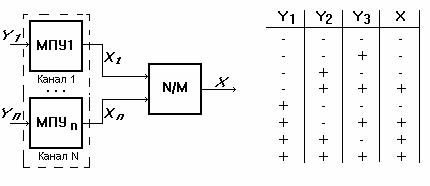


Рис.31

Распространенными методами аппаратурного резервирова-ния являются методы, основанные на мажоритарной обработке и обработке с переключением каналов. При мажоритарной обработ-ке МПС состоит из *п* независимых каналов обработки информации (рис. 31) и остается в рабочем состоянии до тех пор, пока сохра-няют работоспособность *t* из *п* каналов . Например, в системе «2 из З» работоспособное состояние канала определяется из табл. 7, в

178

которой значками «+» и «-» определено соответственно работо-способное 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) неис-правных каналов и их переключения БП (рис.32).

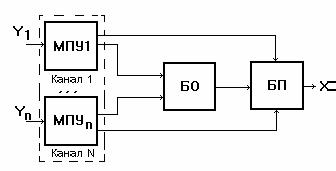


Рис.32

**8.4. Иерархические системы**

Сложные объекты управления (самолеты, космические аппараты, прокатные станы, роторные конвейерные линии и т. п.) представляют собой совокупность взаимосвязанных многорежим-ных управляемых систем, объединенных единой системой управ-ления. Основополагающими принципами, определяющими струк-туру МПС управления подобными объектами, является иерархич-ность, независимость управления по уровням иерархии и инфор-мационная 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 кон-троллера связано с необходимостью запоминания каждого значе-ния управляющего сигнала после останова вычислительного про-цесса. Учитывая, что в распределенной МПС число выходных сигналов невелико, затраты на ЦАП оказываются относительно небольшими.

179

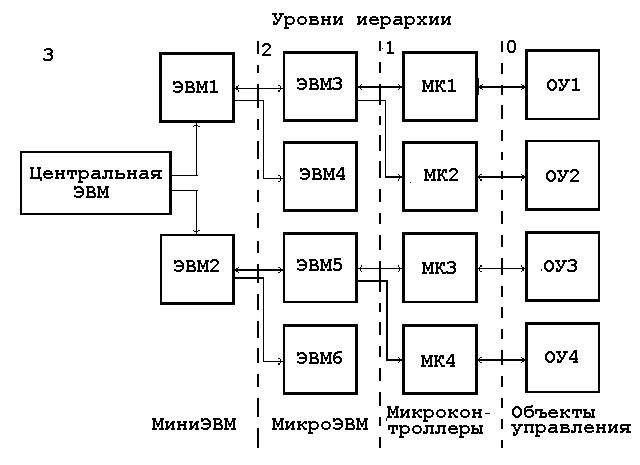


Рис.33

Преобразователи аналоговых сигналов в код позволяют со-прягать микроконтроллер с непрерывными и дискретными датчи-ками, с исполнительными механизмами пропорционального, пози-ционного, интегрирующего и другого действия, а также с различ-ными устройствами дискретного и логического управления .

В микроконтроллере может быть применен микропроцессор как с аппаратным, так и с микропрограммным принципом управ-ления. Аппаратное управление основано на внутреннем микропро-граммировании. При аппаратном управлении система команд мик-ропроцессора является фиксированной. Она реализована во внут-ренних жестких электрических связях в кристалле МП и не может быть изменена разработчиком системы. Микропрограммное

180

управление основано на внешнем микропрограммировании (на-бор команд может быть нефиксированным и изменяться разработ-чиком системы).

Особенностью программного обеспечения микроконтроллера является то, что большая часть его памяти программируется на за-воде - 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 не-больших массивов информации.

Пульт оператора в микроконтроллере используется для уста-новки требуемой конфигурации регулирующего контура, выбора алгоритма управления, контроля значений технологических пере-менных, оперативного вмешательства в процесс управления и дру-гих целей. Программное обеспечение состоит из программ: дис-петчера (координирующего весь вычислительный процесс), рабо-чих, обслуживания пульта и диагностических. Для программиро-вания 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 поня-тие виртуальной (кажущейся) структуры. Виртуальная структура описывает свойства контроллера в традиционных для МПС управ-ления понятиях, основными из которых являются каналы управле-ния, с системной точки зрения эквивалентные отдельному прибору или типовому сочетанию приборов непрерывной системы управ-ления, и конфигурация, определяющая систему связи каналов со входами и выходами контроллера, а также варианты взаимодейст-вия каналов.

По оценкам специалистов, существует ограниченное число (ориентировочно 20 - 25) алгоритмов, комбинация которых позво-ляет автоматизировать управление процессами и объектами прак-тически любой степени сложности. Эти алгоритмы, оформленные в виде библиотеки программ, хранятся в постоянной памяти и мо-гут быть использованы в любом заданном сочетании. Среди про-

181

граммируемых регулирующих микроконтроллеров особое ме-сто занимают однокристальные микроконтроллеры, выпускаемые серийно. По степени универсальности использования их подраз-деляют на специализированные, работающие по жесткой про-грамме, и широкого применения, программа действия которых за-носится во внешнее запоминающее устройство и может изменять-ся самим пользователем или по картам-заказам, составленным пользователем .

Примерами однокристальных перепрограммируемых микро-контроллеров являются контроллеры серии К145. Это цифровые структуры последовательного действия, использующие принцип многоуровневого программирования. Однокристальные микро-контроллеры адаптируются к внешним устройствам как по форма-ту 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 свя-занных объектов. На этом уровне, соответствующем локальному управлению, применяют серийные микроЭВМ многофункцио-нального назначения.

Третий уровень включает управляющие устройства, реализо-ванные на базе мини-ЭВМ, которые координируют работу группы локальных систем.

На четвертом уровне располагается центральная управляю-щая 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

182

связи с объектами управления или измерительными объектами , состоящую из устройств ввода-вывода, которые в общем виде но-сят название контроллеров ввода - вывода (КВВ), внешних уст-ройств, в качестве которых используются датчики и исполнитель-ные 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 ма-гистралью посредством соответствующих интерфейсов. В свою очередь магистраль состоит из шины адресов, шины данных и управляющей шины.

Такая система на практике получила название «общая ши-

на».

В большинстве случаев МПС используют магистрально - модульный принцип построения. Суть этого принципа заключа-ется в том, что отдельные блоки являются функционально закон-ченными модулями со своими встроенными схемами управления, выполненными в виде одного или нескольких кристаллов БИС или СБИС.

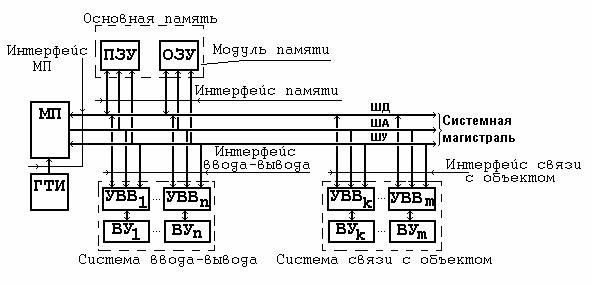


Рис. 34

Межмодульные связи и обмен информацией между моду-лями осуществляются посредством шин (магистралей), к которым

183

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 од-новременного доступа к одной магистрали, является недопусти-мой, так как она приводит к появлению конфликта. Поэтому обмен информацией в магистрально-модульных системах производится, как правило, путем разделения (арбитража) во времени управле-ния модулями системы магистралей.

Особенностью магистрально-модульного принципа по-строения МПС является необходимость информационно - логиче-ской 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 выполне-ния определенных электрических и конструктивных требований при построении интерфейсов.

При построении МПС наибольшее применение получила трехшинная структура, содержащая шину адреса ША, двунаправ-ленную 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 перифе-рийных устройств — внешних запоминающих устройств (ВЗУ) и УВВ.

Модуль памяти ОП (или основная память) МПС содержит постоянное запоминающее устройство ПЗУ и оперативное запо-минающее устройство ОЗУ. В большинстве системах ОП физиче-ски 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

184

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 диа-пазоне целых чисел от 0 до 216 - 1 ( до 65535). Следовательно, об-щее число всех адресов - адресное пространство ОП, составляет 216 = 65 536 = 64К. Эта величина, указываемая в технических ха-рактеристиках, называется размером адресного пространства, вы-раженным числом адресов байт. Иногда ее называют также емко-стью 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 со-блюдаться программистом, так как при его нарушении многие ко-манды будут выполняться с ошибкой.

Чтобы дать более полное представление о МПС, помимо высокой производительности необходимо назвать и другие отли-чительные особенности. Прежде всего это необычные архитектур-

185

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, используе-мый в системе , и организацию памяти, и топологию связи между процессорами, и способ исполнения системой арифметических операций. Эти вопросы освещаются ниже.

1. **Архитектуры с параллельной обработкой данных** 
   * 1966 году М.Флинном был предложен подход к класси-фикации архитектур вычислительных систем. В основу было по-ложено понятие потока, под которым понимается последователь-ность элементов, команд или данных, обрабатываемая процессо-ром. Соответствующая система классификации основана на рас-смотрении числа потоков инструкций и потоков данных. Она со-держит 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 дан-ных. 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 высокопроизводи-тельные системы имеют более одного центрального процессора, однако, каждый из них выполняют несвязанные потоки инструк-ций, что делает такие системы комплексами SIMD-систем, дей-ствующих на разных пространствах данных.

186



Рис. 36

Для увеличения скорости обработки команд и скорости вы-полнения арифметических операций может применяться конвей-ерная 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).

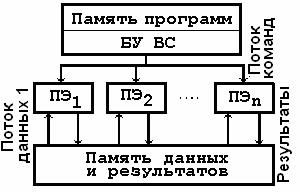


Рис. 37

Определение подразумевает наличие в архитектуре многих процессоров, обрабатывающих один и тот же поток данных. Одна-ко развитие эта архитектура МПС до сих пор не получила.

По ряду признаков к этому классу можно отнести конвей-ерные МПС, однако это не нашло окончательного признания на практике.

187

SIMD - одиночный поток команд и множественный по-ток данных (рис.38). Эти системы обычно имеют большое количе-ство процессоров (от 1024 до 16384), которые могут выполнять одну и ту же инструкцию относительно разных данных в жесткой конфигурации. Единственная инструкция параллельно выполняет-ся над многими элементами данных.

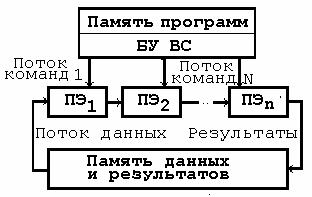


Рис. 38

Примерами SIMD машин являются системы CPP DAP, Gamma II и Quadrics Apemille. Другим подклассом SIMD-систем являются векторные МПС. Они манипулируют массивами сход-ных данных подобно тому , как скалярные системы обрабатывают отдельные элементы таких массивов. Это делается за счет исполь-зования специально сконструированных векторных центральных процессоров. Когда данные обрабатываются посредством вектор-ных модулей, результаты могут быть выданы на один, два или три такта частоты генератора (такт частоты генератора является ос-новным временным параметром системы).

При работе в векторном режиме векторные процессоры об-рабатывают данные практически параллельно, что делает их в не-сколько 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 не-сколько потоков инструкций над различными потоками данных. В отличие от многопроцессорных SISD-систем, упомянутых выше,

188

команды и данные связаны, потому что они представляют раз-личные части одной и той же выполняемой задачи. Например, MIMD-системы могут параллельно выполнять множество подза-дач, с целью сокращения времени выполнения основной задачи.

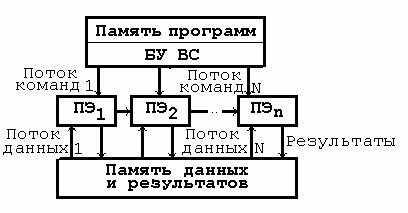


Рис . 39

Наличие большого разнообразия попадающих в данный класс систем, делает классификацию Флинна не полностью адек-ватной . Действительно и четырехпроцессорный SX-5 компании NEC и тысячепроцессорный Cray T3E оба попадают в этот класс. Это заставляет использовать другой подход к классификации, ина-че описывающий классы МПС. Основная идея такого подхода мо-жет состоять, например, в следующем. Считаем, что множествен-ный поток команд может быть обработан двумя способами : либо одним конвейерным устройством обработки, работающем в режи-ме разделения времени для отдельных потоков, либо каждый по-ток обрабатывается своим собственным устройством. Первая воз-можность используется в MIMD-системах, которые обычно назы-вают конвейерными или векторными, вторая – в параллельных системах.

В основе векторных МПС лежит концепция конвейериза-ции, 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-

189

ров, работающих сообща, причем процессоры могут быть как скалярными, так и векторными.

**8.6.1. SMP архитектура**

SMP архитектура (symmetric multiprocessing) - симметрич-

ная многопроцессорная архитектура (рис.40). Главной особенно-стью систем с архитектурой SMP является наличие общей физиче-ской памяти, разделяемой всеми процессорами.

Память является способом передачи сообщений между процессорами, при этом все вычислительные устройства при об-ращении к ней имеют равные права и одну и ту же адресацию для всех ячеек памяти. Поэтому SMP архитектура называется симмет-ричной. Последнее обстоятельство позволяет очень эффективно обмениваться данными с другими вычислительными устройства-ми.

SMP-система строится на основе высокоскоростной сис-

темной шины (SGI PowerPath, Sun Gigaplane, DEC TurboLaser), к

слотам которой подключаются функциональные блоки трех типов: процессоры (ЦП), операционная система (ОП) и подсистема вво-да/вывода (I/O). Для подсоединения к модулям I/O используются уже более медленные шины (PCI, VME64).

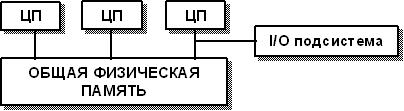


Рис. 40

Наиболее известными SMP-системами являются SMP-cервера 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

190

(в процессе работы ) распределяет процессы по процессорам, но иногда возможна и явная привязка.

Основные преимущества SMP-систем:

* простота и универсальность для программирования. Ар-хитектура SMP не накладывает ограничений на модель програм-мирования, используемую при создании приложения: обычно ис-пользуется модель параллельных ветвей, когда все процессоры ра-ботают абсолютно независимо друг от друга - однако, можно реа-лизовать и модели, использующие межпроцессорный обмен. Ис-пользование 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 ис-пользуют систему охлаждения, основанную на воздушном конди-ционировании, 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 счи-тать их по-настоящему перспективными. Причины плохой мас-штабируемости состоят в том , что в данный момент шина способ-на обрабатывать только одну транзакцию, вследствие чего возни-кают проблемы разрешения конфликтов при одновременном об-ращении нескольких процессоров к одним и тем же областям об-щей физической памяти. Вычислительные элементы начинают друг другу мешать. Когда произойдет такой конфликт, зависит от скорости связи и от количества вычислительных элементов. В на-стоящее 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 произ-водительности при увеличении числа процессоров и числа под-ключаемых пользователей.

191

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 архитекту-ры состоит в том, что память физически разделена. В этом случае система строится из отдельных модулей, содержащих процессор, локальный банк операционной памяти (ОП), два коммуникацион-ных процессора (рутера) или сетевой адаптер, иногда - жесткие диски и/или другие устройства ввода /вывода. Один рутер исполь-зуется для передачи команд, другой - для передачи данных. По су-ти, такие модули представляют собой полнофункциональные сис-темы (рис.41).

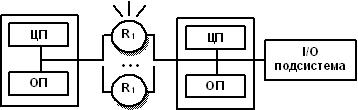


Рис. 41 Доступ к банку ОП из данного модуля имеют только про-

цессоры (ЦП) из этого же модуля. Модули соединяются специаль-ными коммуникационными каналами . Пользователь может опре-делить 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 опера-ционная система (ОС) работает только на управляющей машине (front-end), на каждом отдельном модуле работает сильно урезан-ный вариант ОС, обеспечивающий работу только расположенной в нем ветви параллельного приложения. Во втором варианте на каж-дом модуле работает полноценная UNIX-подобная ОС, устанавли-ваемая отдельно на каждом модуле.

192

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 необ-ходимости в потактовой синхронизации процессоров. Практически все рекорды по производительности на сегодняшний день уста-навливаются 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 меж-процессорного обмена, поскольку нет общей среды для хранения данных, предназначенных для обмена между процессорами. Тре-буется специальная техника программирования для реализации обмена сообщениями между процессорами;
* каждый процессор может использовать только ограничен-ный объем локального банка памяти;
* вследствие указанных архитектурных недостатков требу-ются значительные усилия для того, чтобы максимально использо-вать системные ресурсы. Именно этим определяется высокая цена программного обеспечения для массивно-параллельных систем с раздельной памятью.

Системами с раздельной памятью являются суперкомпью-

теры МВС-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 производительно-стью 1200 Мфлопс/с (CRAY T3E-1200), способны масштабиро-ваться до 2048 процессоров.

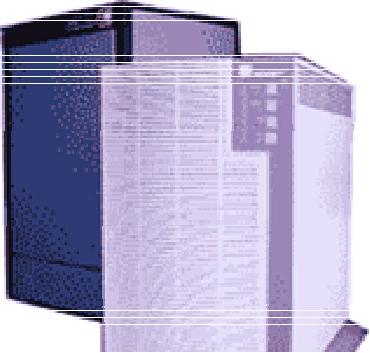
**8.6.3. Гибридная архитектура**

Гибридная архитектура (NUMA - nonuniform memory access) характеризуется неоднородным доступом к памяти. Она воплощает в себе удобства систем с общей памятью и относитель-ную дешевизну систем с раздельной памятью. Суть этой архитек-туры поясним на примере архитектуры сервера NUMA Q-2000, разработанного фирмами IBM и Sequent в 1999 году. Он имеет

193

особую организацию памяти, а именно: память является физи-чески распределенной по различным частям системы, но логиче-ски разделяемой, так что пользователь видит единое адресное про-

странство. 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 (cимметрич-ная многопроцессорная архитектура) узлы.

Структурная схема системы с гибридной сетью представле-на на рис. 42. Она содержит четыре процессора, которые связыва-ются между собой при помощи кроссбара в рамках одного SMP узла. Узлы связаны сетью типа "бабочка" (Butterfly):

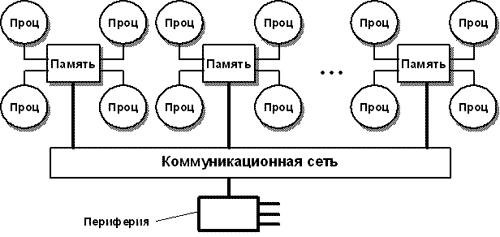


Рис. 42

194

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 - коге-рентный кэш, создав так называемую архитектуру cc-NUMA (Cache Coherent Non-Uniform Memory Access), которая расшифро-

вывается как "неоднородный доступ к памяти с обеспечением ко-герентности кэшей". Он ее реализовал на системах типа Origin.

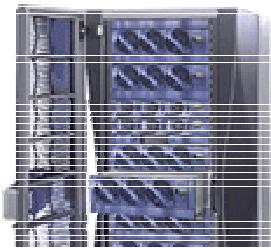
Понятие когерентности кэшей описывает тот факт , что все центральные процессоры получают одинаковые значения одних и тех же переменных в любой момент времени. Действительно, по-скольку кэш-память принадлежит отдельной системе, а не всей многопроцессорной системе в целом , данные, попадающие в кэш одной МПС, могут быть недоступны другой. Чтобы избежать это-го, следует провести синхронизацию информации, хранящейся в кэш-памяти процессоров.

Для обеспечения подобной когерентности кэшей сущест-вуют несколько возможностей:

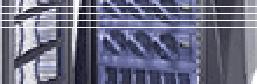
* использовать механизм отслеживания шинных запросов (snoopy bus protocol), в котором кэши отслеживают переменные, передаваемые к любому из центральных процессоров и, при необ-ходимости, модифицируют собственные копии таких переменных;
* выделять специальную часть памяти, отвечающую за от-слеживание достоверности всех используемых копий перемен-ных.

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 динамиче-ского "подразделения" системы, когда отдель-



195

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 яв-ляется наличие специальных векторно-конвейерных процессоров, в которых предусмотрены команды однотипной обработки векто-ров независимых данных, эффективно выполняющиеся на конвей-ерных функциональных устройствах. Как правило, несколько та-ких процессоров (от 1 до16) работают одновременно с общей па-мятью (аналогично SMP) в рамках многопроцессорных конфигу-раций. Несколько таких узлов могут быть объединены с помощью коммутатора (аналогично MPP). Поскольку передача данных в векторном формате осуществляется намного быстрее, чем в ска-лярном (максимальная скорость может составлять 64 Гб/с, что на 2 порядка быстрее , чем в скалярных системах), то проблема взаимо-действия между потоками данных при распараллеливании стано-вится 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 яв-ляются:

* CRAY SV-2, SMP архитектура. Пиковая производительность системы в стандартной конфигурации может составлять десятки те-рафлопс.
* NEC SX-6, NUMA архитектура. Пиковая производительность системы может достигать 8 Тфлопс, производительность 1 процессора составляет 8 Гфлопс. Система масштабируется до 128 узлов.



196

**-** Fujitsu-VPP5000 (vector parallel processing) ), MPP

архитектура. Производительность 1 процессора составляет 9.6 Гфлопс, пиковая производительность системы мо-жет 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 техноло-гий 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 работоспо-собности становится очевидным в случае сбоя какого-либо узла: при этом другой узел кластера может взять на себя нагрузку неис-правного узла, и пользователи не заметят прерывания в доступе. Возможности масштабируемости кластеров позволяют многократ-но увеличивать производительность приложений для большего числа пользователей . технологий на базе шинной архитектуры или коммутатора. Такие суперкомпьютерные системы являются самы-ми дешевыми, поскольку собираются на базе стандартных ком-плектующих элементов ("off the shelf"), процессоров, коммутато-ров, дисков и внешних устройств.

Кластеризация может быть осуществлена на разных уров-нях компьютерной системы, включая аппаратное обеспечение, операционные системы, программы-утилиты, системы управления и приложения. Чем больше уровней системы объединены кластер-ной технологией, тем выше надежность, масштабируемость и управляемость кластера.

Кластеры условно делятся на классы:

197

Класс I. Класс строится целиком из стандартных дета-лей, которые продают многие продавцы компьютерных компонент (низкие цены, простое обслуживание, аппаратные компоненты доступны из различных источников).

Класс II. Система имеет эксклюзивные или не широко рас-пространенные детали. Этим можно достичь очень хорошей про-изводительности, 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 мо-жет иметь такие свойства или функции, которые выходят за рамки перечисленных типов. Более того, при конфигурировании большо-го кластера, используемого как система общего назначения, при-ходится выделять блоки, выполняющие все перечисленные функ-ции.

**Кластеры для высокопроизводительных вычислений**

предназначены для параллельных расчётов . Эти кластеры обычно собраны из большого числа компьютеров. Разработка таких кла-стеров является сложным процессом, требующим на каждом шаге аккуратных согласований таких вопросов как инсталляция, экс-плуатация и одновременное управление большим числом компью-теров, технические требования параллельного и высокопроизводи-тельного доступа к одному и тому же системному файлу (или фай-лам) и межпроцессорная связь между узлами и координация рабо-ты 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 обеспеченияединого интерфейса к ряду ресурсов, которые могут со временем

198

произвольно наращиваться (или сокращаться) в размере. Наи-более 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 ресурса-ми проекта Earth and Space Sciences. Проектно -конструкторские работы над кластером быстро превратились в то, что известно сей-час под названием проект Beowulf.

Проект стал основой общего подхода к построению парал-лельных кластерных компьютеров и описывает многопроцессор-ную 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 под-чинённых узлов (вычислительных узлов), соединённых посредст-вом стандартной компьютерной сети. Система строится с исполь-зованием стандартных аппаратных компонент, таких как ПК, за-пускаемых под Linux, стандартных сетевых адаптеров (например, Ethernet) и коммутаторов. Нет особого программного пакета, на-зываемого «Beowulf». Вместо этого имеется несколько кусков программного обеспечения, которые многие пользователи нашли пригодными для построения кластеров Beowulf. Beowulf исполь-зует такие программные продукты как операционную систему Linux, системы передачи сообщений PVM, MPI, системы управле-ния очередями заданий и другие стандартные продукты. Сервер-ный узел контролирует весь кластер и обслуживает файлы, на-правляемые к клиентским узлам.

При разработке кластерных систем существует ряд про-блем, среди которых можно выделить следующие.

Архитектура кластерной системы (способ соединения про-цессоров друг с другом ) в большей степени определяет ее произ-водительность, чем тип используемых в ней процессоров. Крити-ческим параметром, влияющим на величину производительности такой системы, является расстояние между процессорами. Так, со-

199

единив вместе 10 персональных компьютеров, мы получим сис-тему для проведения высокопроизводительных вычислений, про-блема, однако, будет состоять в нахождении наиболее эффектив-ного способа соединения стандартных средств друг с другом, по-скольку при увеличении производительности каждого процессора

* 10 раз производительность системы в целом в 10 раз не увели-чится.

Рассмотрим для примера задачу построения симметричной 16-ти процессорной системы, в которой все процессоры были бы равноправны. Наиболее естественным представляется соединение

* виде плоской решетки, где внешние концы используются для подсоединения внешних устройств (рис.43).

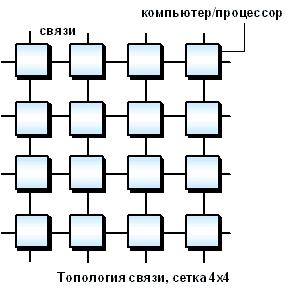


Рис. 43

При таком типе соединения максимальное расстояние меж-ду процессорами окажется равным 6 (количество связей между процессорами, отделяющих самый ближний процессор от самого дальнего). Теория же показывает, что если в системе максималь-ное расстояние между процессорами больше 4, то такая система не может работать эффективно. Поэтому, при соединении 16 процес-соров друг с другом плоская схема является не эффективной.

Для получения более компактной конфигурации необходи-мо решить задачу о нахождении фигуры, имеющей максимальный

200

объем при минимальной площади поверхности. В трехмерном пространстве таким свойством обладает шар. Но поскольку нам необходимо построить узловую систему, то вместо шара прихо-дится использовать куб (если число процессоров равно 8) или ги-перкуб, если число процессоров больше 8. Размерность гиперкуба будет определяться в зависимости от числа процессоров, которые необходимо соединить. Так, для соединения 16 процессоров по-требуется 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). Ар-хитектура гиперкуба является второй по эффективности, но самой наглядной.

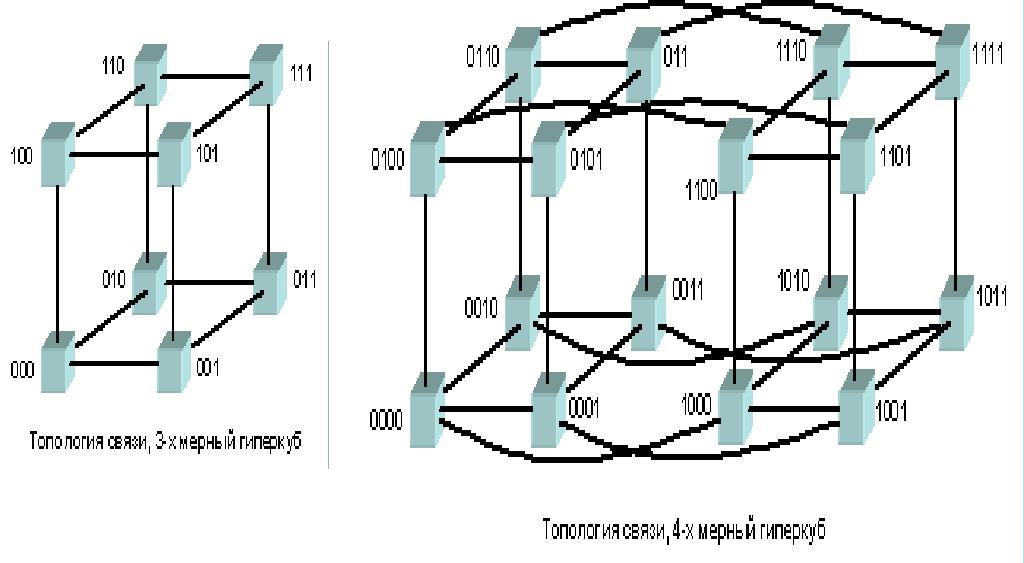


Рис. 44

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 внут-ренние узлы дерева скомпонованы во внутреннюю сеть. Поддере-

201

вья могут общаться между собой, не затрагивая более высоких уровней сети.

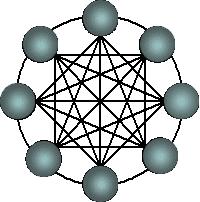


Рис. 45

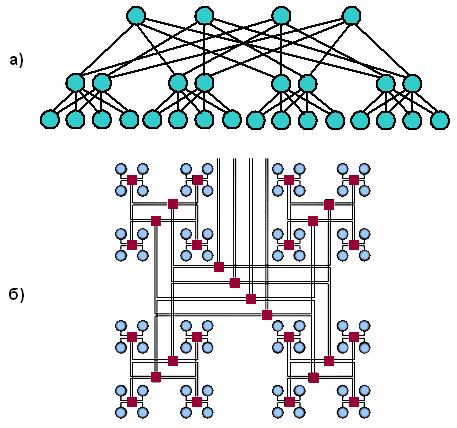


Рис. 46

Поскольку способ соединения процессоров друг с другом больше влияет на производительность кластера, чем тип исполь-зуемых в ней процессоров, то может оказаться более рентабель-

202

ным создать систему из большего числа дешевых компьютеров, чем из меньшего числа дорогих.

В кластерах, как правило, используются операционные сис-темы, стандартные для рабочих станций, чаще всего, свободно распространяемые **-** Linux**,** FreeBSD**,** вместе со специальными средствами поддержки параллельного программирования и балан-сировки нагрузки.

.

**8.6.6. Транспьютеры**

Транспьютер – это слово , производное от слов транзистор и компьютер . Транспьютер - это микроэлектронный прибор, объе-диняющий на одном кристалле микропроцессор, быструю память, интерфейс внешней памяти и каналы ввода-вывода (линки), пред-назначенные для подключения аналогичных приборов. Прибор спроектирован таким образом, чтобы максимально облегчить по-строение параллельных вычислительных систем. При соединении транспьютерных элементов между собой требуется минимальное число дополнительных интегральных схем. Связь между транс-пьютерами 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 вы-пуска 32 – разрядного транспьютера Т414 с фоннеймановской КСНК – архитектурой (рис. 47).

Взаимодействие каждого транспьютера с другими транс-пьютерами и периферийными устройствами осуществляется по-средством 4-х коммуникационных каналов связи, имеющихся в составе БИС. Для передачи сообщений из внутренней и внекри-стальной локальной памяти по последним каналам применяется механизм блочных ПДП-пересылок. Интерфейсы связи и процес-сор работают одновременно, что приводит лишь к незначительной потери производительности процессора. Использование прямых последовательных коммуникационных каналов делает ненужным арбитраж приоритетов и исключает проблемы, связанные с пропу-скной способностью шин и их перегрузкой при введении в систе-му новых процессоров.

203

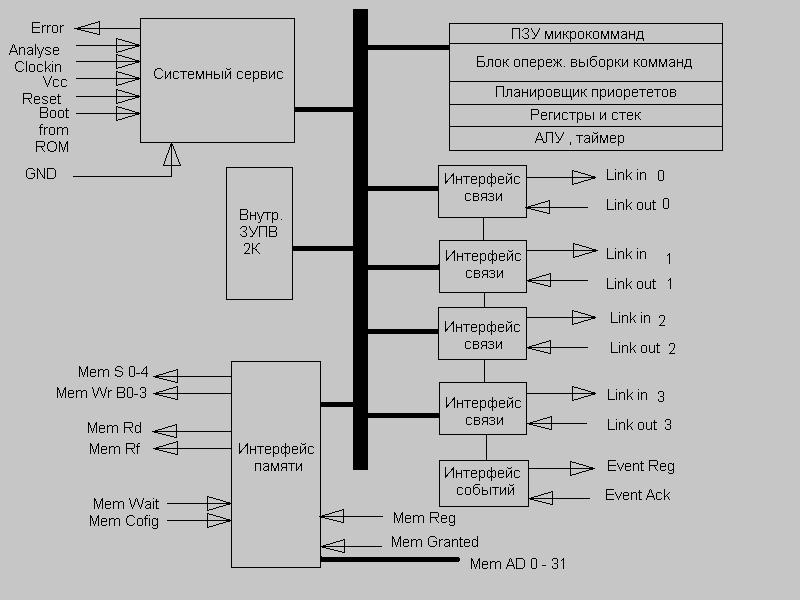


Рис.47.

Каждый исследуемый канал состоит из двух частей, слу-жащих для передачи информации в противоположных направле-ниях. 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. По-сле передачи байта данных пославший его транспьютер ожидает получения двухбитового подтверждающего сигнала, указывающе-го на то, что принимающий транспьютер готов к дальнейшему приему информации. Возможен обмен информации между незави-

204

симотактируемыми системами, если частоты тактирования одинаковы.

Для сопряжения последовательных каналов транспьютера с нетранспьютерными устройствами и интерфейсами связи преду-смотрен ряд адаптерных схем. Контроллеры переферийных уст-ройств также могут быть присоединены к имеющейся памяти, по-средством которой возможно их обращение ко всему пространству памяти.

Система прерываний в традиционном смысле этого понятия отсутствует, но имеют место аналогичные средства, реализован-ные в виде двух уровнего приоритета, присваиваемых процессам, ожидающим приема по последовательным каналам и запросному входу события.

Транспьютер может быть использован в качестве отдельно-го самостоятельного устройства, обеспечивающего производи-тельность 10 млн. оп/ с. При этом для программирования использу-ется широкий набор стандартных высокоуровневых языков, т.к. архитектура транспьютера ориентирована на эффективное приме-нение компиляции. Для полной реализации возможностей объеди-нения транспьютеров в сети или матрицы при построении высоко-производительных систем целесобразно применять язык Окками, позволяющий максимальным образом использовать свойства транспьютеров, ориентированные на распараллеливание обработ-ки.

Аппаратные средства транспьютера прямым образом ори-ентированы на реализацию параллельной обработки и соответст-вующих информационных пересылок (рис. 48).

Планировщик поволяет одновременно выполнять любое количество параллельных процессов, между которыми распреде-ляется процессорное время. Время переключения процессов со-ставляет менее 1 мкс, а обмен информации между процессами осуществляется посредством блочных передач ввод-вывод памяти.

Активные процессы, ожидающие выполнения, содержатся в связанной очереди рабочих просранств, которая реализована с по-мощью двух регистров, один указывает на первый процесс в оче-реди, а второй указывает на последний процесс. Как только вы-полнение процесса становится возможным, указатель команды за-

205

поминается в его рабочем пространстве, и из очереди берется для выполнения следующий процесс.

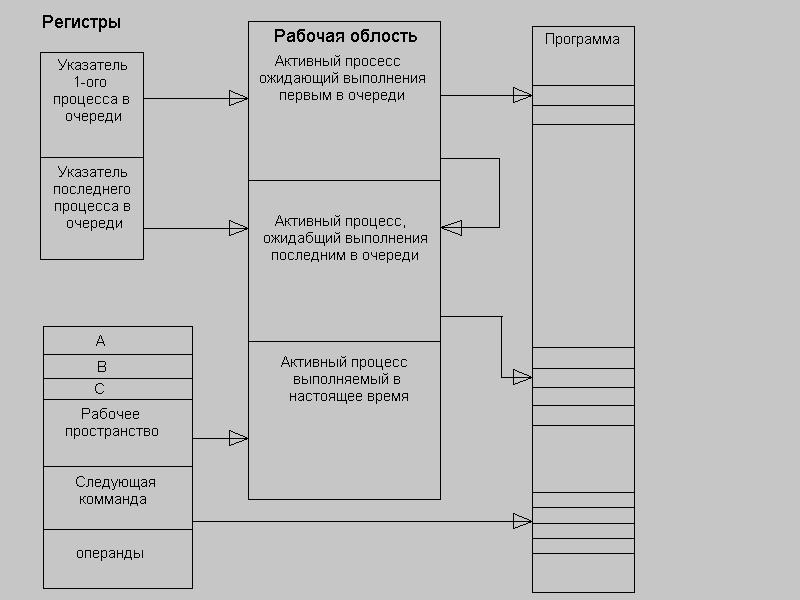


Рис. 48

Транспьютер обеспечивает двухуровневую систему при-оритетов. 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 оп-рделенной периодичностью на интервалы времени, чтобы процес-сорное время равномерно распределялось междузадачами, обра-ботка которых требует больших временных затрат. Продолжи-тельность периода квантования составляет 4096 периодов входно-го текстового сигнала, имеющего частоту 5 МГц, т.е. примерно 800 мкс. Не должна иметь место ситуация , когда весь период кван-тов был бы занят выполнением процессов высокого уровня.

206

Подготовка процесса к вводу или выводу заключается в загрузке в вычислительный стек:

* индентификатора канала;
* количества подлежащих пересылке байтов;
* указателя буфера.

Передача сообщения по внутренему каналу осуществляется путем занесения идентификатора первого процесса, оказавшегося готовым в слово канала и занесения указателя в рабочее простран-ство. После этого процессор выполняет очередной процесс из пла-нирующей очереди.

Когда оказывается готовым второй процесс, использующий этот же канал, происходит перепись сообщения, ожидающий про-цесс добавляется к очереди активных процессов и производится начальная установка канала в пустое состояние.

Когда сообщение пересылается по внешнему каналу, про-цессор возлагает эту передачу на автономный интерфей канала и исключает процесс из плана передач. Есле интерфейс канала реа-лизует передачу сообщения посредством прямого доступа к памя-ти, процессор вновь вводит ожидающий процесс в расписание пе-ресылок. В то время когда имет место внешняя пересылка сооб-щения, 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 произво-дится инициализация этих трех регистров и занесение указателя команды в рабочеепространство процесса. После того как оба про-цесса инициализируют свои канальные интерфейсы, происходит перепись сообщений и оба интерфейса ставят процесс в конец ло-кальной очереди активных процессов.

Способ, последством которого транспьютерное семейство реализует параллельную обработку при использованиинескольких транспьютерных СБИС, определяется организацией последова-тельных каналов и Оккам- каналов. Каждый последовательный ка-нал позволяет реализовать два Оккам-канала, по одному для каж-

207

дого из двух направлений передачи. Пока ЦПУ занят обработ-кой, обмен информацией может происходить по всем четырем ка-налам.

При времени цикла транспьютера Т414 50 нс на 4-е канала одновременно работающих в обоих направлениях со скоростью 100Мбит/с с использованием внутренной памяти затрагивается 8% производительности ЦПУ.

Сообщения пересылаются в виде последовательностей бай-тов. Каждый 8-битовый байт предваряется двумя единичными би-тами 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 на-чинает передаваться после получения двух заголовочных байтов.

Необходимо отметить, что архитектура транспьютера включает решение проблемы соединений. Для решения этой про-блемы INMOS мог бы добавить несколько дополнительных ком-мукационных 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 фи-зический 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 кана-лами и их число определяется решаемой программой . С помощью чипов коммуникации, отвечающих за эффективную доставку со-общений по указанному назначению, виртуальные каналы могут запускаться между транспьютерами, которые не соединены на-прямую физическими коммуникационными каналами.

Когда осуществляется программирование какой-либо зада-чи, то необходимо однозначно определить, какой физический коммуникационный канал закрепляется за каждым каналом пере-дачи сообщений и может потребоваться несколько таких одно-значных определений для приема сообщения от отдаленного транспьютера. Это делает программу зависимой от точной тополо-

208

гии конкретной сети, следовательно, она становится неприло-жимой к сети с любой другой топологией.

В чипе два процесса, расположенные в произвольных мес-тах сети. Они могут использовать один и тот же программный ка-нал для связи. Аппаратура же сама упорядочит маршрут передачи сообщения. С этим улучшением, программы могут быть независи-мыми 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 по-очередной передачи пакетов от нескольких сообщений через один физический коммуникационный канал.

Физические коммуникационные каналы чипа почти впятеро быстрее (100 Мбит/сек полный дуплекс), чем транспьютерные фи-зические коммуникационные каналы. Кроме того, системный па-кет обеспечивает более эффективное использование канала, по-этому производительность коммуникационного канала будет улучшена немотря на сложность мультиплексирования.

Чип разделяет сообщения произвольной длины на последо-вательность 32-байтовых пакетов, в отличие от существующего транспьютера, который посылает сообщения байт за байтом. Сме-шанные 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 меркетом ЕОР (End-Of- Packet - конец пакета), за исключением последнего пакета, который имеет маркер ЕОМ (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 па-кет).

Для сохранения синхронизации передачи сообщений, при-емник должен подтверждать получение каждого пакета; подтвер-ждение пакета - это пакет, не содержащий данных. Каждое вирту-

209

альное соединение содержит два виртуальных канала , один для передачи пакетов данных вовне, другой для приема подтвержде-ний. Процесс посылки пакетов сообщений не может продолжаться до тех пор, пока переддддача последнего пакета не будет подтвер-ждена. Подтверждение посылается тогда, когда принимается пер-вый байт пакета, оно разрешает непрерывную передачу сообще-ний, если приемник готов к вводу информации.

Запросы на передачу ставятся в очередь по каждому физи-ческому коммуникационному каналу связи, так что от МП не тре-буется 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 суще-ствующие транспьютеры буферизуют 1 байт в аппаратном регист-ре. В чипе процесс буферизации будет осуществляться в память, что предпочтительнее буферизации в регистр, поэтому в случае достаточного объема и производительности памяти на одном про-цессоре может быть образовано любое число виртуальных кана-лов.

Для полной реализации концепции виртуальных каналов, требуется дополнительный чип переключения марщрутов С104 (коммутатор) (рис. 49).

Чип С104 является полным чипом обмена коммутируемыми пакетами, аналогичным по принципу работы телефонной сети ор-ганического доступа для обмена деловой информацией. Чип С104 содержит 32 физических коммуникационных транспьютерных ка-нала и может коммутировать сообщение от одного из 32 чипов к любым другим чипам.

Чип С104 содержит коммутатор 32×32 для соединения лю-бого из его физических коммуникационных каналов с любым дру-гим, и чудесно простую логику для подключения адресата для ка-ждого принятого сообщения. Чипу не требуется внутреннее CPU или значительное количество памяти благодаря схеме коммута-ции, примененной разработчиками INMOSa.

210

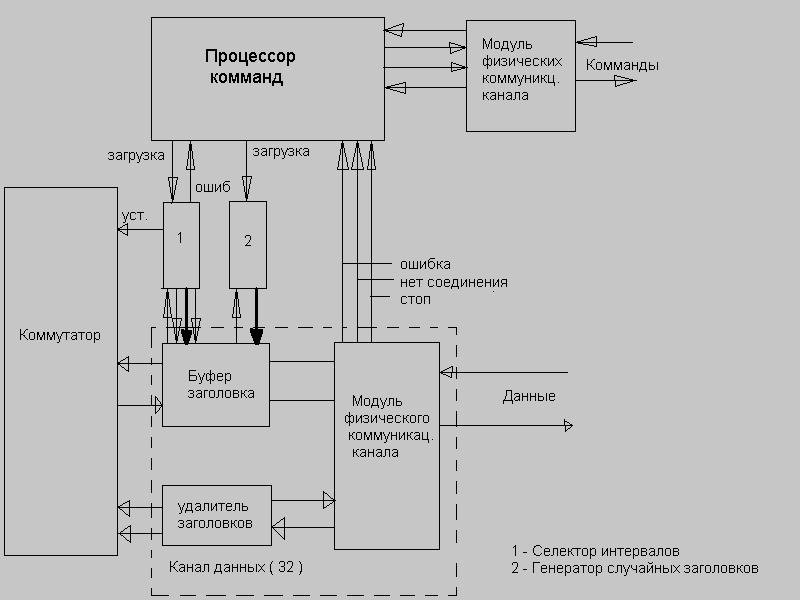


Рис.49

В большинстве сетей коммутации пакетов каждый пере-ключатель маршрута принимает полный пакет и сохраняет его в буфере, декодирует адресную информацию пакета, и пересылает пакет к следующему узлу назначения. Эта схема не очень подхо-дит для высокопроизводительных транспьютерных сетей , по-скольку она вносит относительно большую задержку между пере-дачей пакета и его прибыванием в пункт назначения (т.е.имеет большое время ожидания), к тому же требует от чипа коммутации С104 наличия промежуточной буферной памяти.

INMOS выбрал вместо этой схемы решение, называемое червячным каналом маршрутизации, в котором читается только

211

заголовок пакета и дешифруется его адрес. Затем, если требуе-мый 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 сред-него размера, заголовок пакета может вообще не появиться от пе-редающего абонента полностью, он достигнет пункта назначения еще до того, как абонент завершит его передачу. В результате, от-крывается сквозной канал от передающего абонента к принимаю-щему, пока данные передаются непрерывно, и который закрыва-ется, как только конец пакета "протянется" через канал. Аналогия

* червячным каналом была выбрана из наблюдения за червяком, прокладывающим канал в песке, который закрывается снова за его хвостом.
  + 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
* очень малым временем ожидания, и он подразумевает, что чипу С104 нужен только 1-2 байтовый буфер для заголовков вместо бу-фера под целый пакет.

Червячный канал маршрутизации в общем невидим для со-общений посылающих и принимающих абонентов, поскольку он создается на аппаратном уровне ниже механизма синхронизации посылки/подтверждения.

Полная маршрутная стратегия требует решения алгоритма маршрутизации пакета, опираясь на адрес, содержащийся в его за-головке, т.е. какой канал будет открыт для передачи пакета. Алго-ритмы маршрутизации являются хорошо изученной областью бла-годаря их колоссальной важности для индустрии передачи сооб-щений. Хороший алгоритм маршрутизации должен быть завер-шенным (т.е. он должен гарантировать, что каждый переданный пакет в результате достигнет абонента), не содержать в себе взаи-моблокировок (dead-lock), должен быть оптимальным (т.е. пакеты

212

должны следовать по кратчайшим маршрутам) и малоизбыточ-ным (т.е. пакетыдолжны 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 маршрути-зации должен быть простым. Простота вдвойне важна, когда вы используете быстрый маршрутизатор червячного канала, во избе-жание ухудшения времени ожидания сети. Не существует алго-ритмов, полностью удовлетворяющих всем этим требованиям.

Одним стандартным решением является таблица маршрути-зации, 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 реше-нием является побитное разрушение, в котором каждый узел сети, которого достигает сообщение, "глотает" один бит из заголовка сообщения; этот алгоритм является завершенным, дешевым, быст-рым 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 завер-шенным, свободным от взаимоблокировок, недорогим, быстрым и масштабируемым , могущим быть близким к оптимальному или многосторонним, в зависимости от того, как он используется. Ка-ждый транспьютер (или другой узел назначения, такой, как межсе-тевой шлюз или переферийный чип) маркируется номером, так сеть из N транспьютеров будет маркирована 0,1,2... N-1. Этот но-мер 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-

213

тервалы не перекрываются и пронумерованы так, что каждый заголовок попадает только в один интервал.

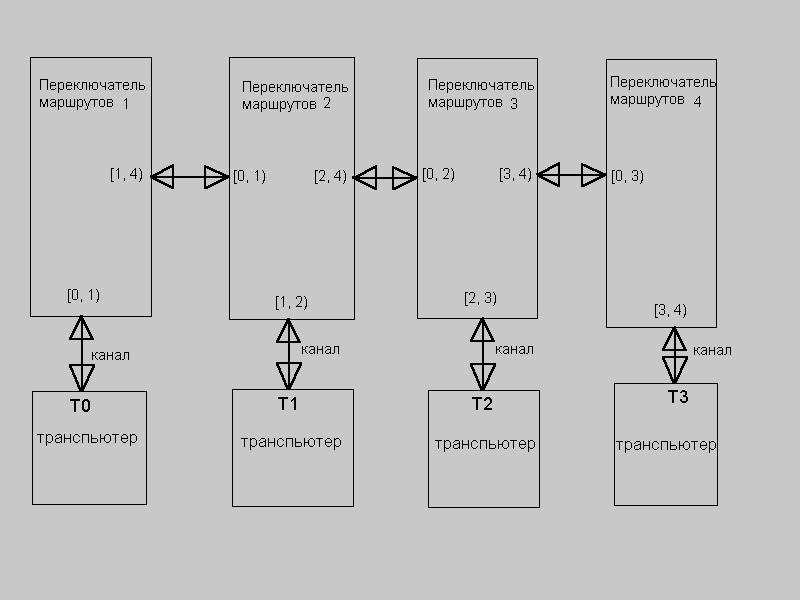


Рис. 50

В результате каждый интервал говорит: "В этом направле-нии лежит часть сети, содержащая все указанные в интервале но-мера процессоров". Для разработчика аппаратуры привлекатель-ность метода интервальной маршрутизации заключается в том, что он может быть применен для маршрутизации пакета с использова-нием одной единственной операции сравнения. Так логика чипа С104 состоит чуть больше, чем из пары компараторов и инте-гральных регистров на каждый физический коммуникационный канал. Чип С104 может использовать одноили двухбайтовые заго-ловки, поэтому в одиночной сети может быть маршрутизировано до 65536 транспьютеров.

214

Хотя схема интервальной маршрутизации может быть гарантирована от взаимоблокировок, она может стать жертвой го-рячих точек. Когда слишком много сообщений маршрутизировано через один узел сети, большинство доставляемых пакетов остано-вится на непредсказуемый интервал времени, решающим образом снижая скорость прохождения через узел, такой узел сети называ-ется горячей точкой. Эта проблема сильно отличается от взаимо-блокировки, которая является логической ошибкой и результат ко-торой бесконечное ожидание сообщения, это скорее свойство то-пологии сети и алгоритма маршрутизации, чем применения. Для любой топологии сети, как вы можете догадаться, вероятно най-дется алгоритм применения, который может порождать горячие точки.

Можно избежать горячих точек равномерным распределе-нием сетевого трафика, и можно запрограмировать чип С104 для выполнения этого распределения. Маршрутизация становится двухфазным процессом , в первой фазе которого пакет посылается в случайно выбранный узел сети, а из этого узла сообщение пося-лается к его пункту конечного назначения - схема называется уни-версальная маршрутизация.

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 производитель-ность сети может ухудшиться на порядки, и фактор дешевизны се-ти перестает играть роль вследствие непригодности сети для рабо-ты. 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 вызы-вает наибольший интерес, процессор содержит ряд других улуч-шений для облегчения жизни разработчиков операционной систе-мы и для улучшения поддержки языков, таких как С и АДА.

Подобно существующему Т800, чип будет иметь встроен-ный сопроцессор плавающей точки (FPU) и четыре физических

215

коммуникационных канала. Ядро процессора будет иметь встроенное кеширование и контроллер DRAM, который будет поддерживать адресные режимы статической памяти. Эти воз-можности в счетании сулучшеной технологией изготовления дают перспективу достижения пиковой производительности в 100 млн. операций в секунду и 20 млн. операций с плавающей точкой в се-кунду. 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. МВС – 1000**

МВС-1000 — система 3-го поколения, разработанная в Рос-сии фирмой «Квант». Она основана на использовании микропро-цессоров Alpha 21164 (разработка фирмы DEC-Compaq; выпуска-ется 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 вторично-го электросилового питания и вентиляцией располагается в стой-ках размером 550x650x2200 мм3 промышленного стандарта ; вес заполненной стойки — 220 кг, потребляемая мощность до 4 кВт. Система МВС-1000 с производительностью до 1 TFLOPS состоит из 8 стоек (512 узлов).

Система реализуется в двух модификациях: на базе «транс-пьютероподобного» связного микропроцессора TMS320C44 (фир-ма Texas Instruments), имеющего 4 канала с пропускной способно-стью каждого — 20 Мбайт/с, либо на базе связного микропроцес-

сора SHARC ADSP 21060 (фирма Analog Devices), имеющего 6

внешних каналов с пропускной способностью каждого — 40 Мбайт/с.

Исполнение МВС-1000К отличается использованием для межпроцессорного обмена коммутационной сети MYRINET (фир-ма Myricom, США) с пропускной способностью канала в дуплекс-ном режиме 2x160 Мбайт/с. Кроме того, предусмотрено подклю-

216

чение к каждому процессору памяти на жестком диске с объе-мом 2-9 Гбайт. В стандартной стойке располагается до 64 процес-соров 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 двухмерного тора (для 4-линковых узлов). Структурный модуль (рис .51) состоит из 16 вычислитель-ных модулей (ВМ ), образующих матрицу 4x4. При этом четыре угловых элемента матрицы соединяются через транспьютерные линки по диагонали попарно. Оставшиеся 12 линков преднзанача-ются 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 ис-ходной матрице 4x4).

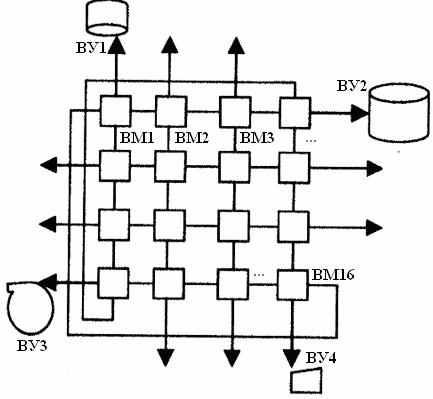


Рис .51

Конструктивным образованием МВС-1000 является базо-вый вычислительный блок, содержащий 32 вычислительных мо-дуля (рис.52). Максимальная длина пути между любыми между любыми из 32 вычислительных модулей равна пяти, как в булев-ском гиперкубе. При этом число свободных линков после ком-плектации блока составляет 16, что позволяет продолжить проце-

217

дуру объединения. Возможна схема объединения двух базовых блоков в 64-процессорную систему, которая приведена на рис. 53.

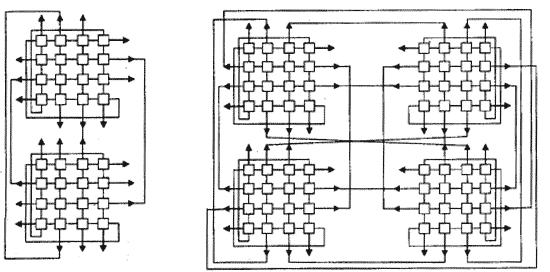


Рис. 52 Рис.53

Для управления массивом процессоров и внешними устрой-ствами, а также для доступа к системе извне используется так на-зываемый 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 операци-

онной 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 вычисли-тельного модуля используются не специализированные ЭВМ, предназначенные только для применения в качестве деталей су-перкомпьютерной установки, а обычные, универсальные персо-нальные компьютеры. Соответственно, и в качестве коммуникаци-онной аппаратуры используются не специализированные «транс-пьютероподобные» процессоры, а обычные сетевые платы и ком-мутаторы, применяемые для построения офисных локальных се-тей. Такой подход стал не только возможным, но и единственно

218

оправданным, по мере совершенствования коммуникационной аппаратуры общего назначения, в первую очередь, с появлением современных сетевых коммутаторов.

В качестве базовой ОС узла используется Linux, что являет-ся, фактически, общепринятым мировым стандартом для построе-ния систем такого класса. Это позволило многократно расширить и упростить, по сравнению с ранними версиями МВС-1000, адап-тацию самого разнообразного программного обеспечения, как сво-бодно распространяемого, так и коммерческого.

**8.6.8. Молекулярные компьютеры.**

Ученые из Калифорнийского университета в Лос-Анджелесе, работающие в составе группы Hewlett-Packard, заяви-ли о создании молекулярных переключателей, которые способны функционировать как элементы памяти.

Над созданием систем нового поколения исследователь-ские 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 чрез-вычайно 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 ин-формации открывает новые физические возможности повышения эффективности вычислительных систем. Это объясняется сущест-венным уменьшением потерь энергии на выполнение одной логи-ческой операции, массовым параллелизмом, высокой плотностью записи информации вплоть до 1 бита на несколько атомов.

Существует огромное число молекулярных систем, имею-щих несколько устойчивых состояний (электрических, магнитных, оптических, химических), в которых можно осуществлять перехо-

219

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, сле-довательно, позволяют создавать устройства с плотностью записи до 1012 бит/см2. При этом, быстродействие молекулярных элемен-тов определяется временами электронных переходов и может дос-тигать величин порядка 109- 1012 оп/сек. Энергопотребление внут-римолекулярных 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 пер-вых методов решения этой задачи была технология пленок Лен-гмюра-Блоджетт.

Первые двоичные молекулярные элементы были основаны на изменении состояния молекулярного ансамбля при воздействии на него электрического сигнала. Одним из таких элементов, явля-ется молекулярная цепочка, представляющая собой чередование потенциальных ям и барьеров. В такой цепочке возможен эффект резонансного туннелирования. При совпадении энергии входяще-го в цепочку свободного электрона с одним из уровней в потенци-альной яме вероятность прохождения электрона через цепочку равна 1. При подаче потенциала на одну из молекул, соответст-вующих квантовой яме уровни смещаются, условия резонанса на-рушаются и цепочка становится непроводящей. На основе эффекта резонансного туннелирования можно реализовать различные ло-гические элементы. Для создания таких элементов используются молекулы порфиринов, соединенные углеводородными мостика-ми.

Для управления состоянием отдельной молекулы приме-няют зонд туннельного микроскопа, который можно использовать для исследований при разработке и создании молекулярного ком-пьютера.

Архитектура параллельных вычислительных систем, тра-диционно используемая в разрабатываемой вычислительной тех-нике, реализуется на основе твердотельной микроэлектронной

220

элементной базы, но совершенно очевидно, что такая архитек-тура вполне подходит для молекулярных систем. Идея параллель-ных вычислений открывает путь к решению двух принципиальных проблем создания молекулярных вычислительных устройств. С одной стороны решается проблема адресации, т.к. отпадает необ-ходимость контакта к каждому логическому молекулярному эле-менту. С другой стороны, многие типы параллельных вычисли-тельных систем устойчивы к выходу из строя определенного числа логических элементов. Это происходит в результате технологиче-ских дефектов молекулярных слоев, которые не приводят к выходу из строя всей вычислительной системы, а лишь ослабляют интен-сивность выходного сигнала.

В качестве вычислительных систем параллельного дейст-вия представляется перспективным использование таких природ-ных биологических объектов, как бактериородопсин, родопсин, фитохром, ряд ферментов и белков, а также природные нейрост-руктуры. При использовании молекулярных структур необходимо решить такую очень важную проблему, как стыковка биологиче-ских молекулярных объектов с макроскопическими опто- и микро-электронными устройствами ввода-вывода информации. В качест-ве основного микроэлектронного объекта стыковки элементов па-мяти с устройствами управления можно рассматривать полевой МДП-транзистор, затвор которого связан с молекулярным фраг-ментом памяти. Очевидно, что при площади затвора транзистора 1 мкм2 будет восприниматься суммарный отклик от большого чис-ла молекул (103-104).

Проблему по снятию сигнала с такого количества молекул можно решать различными методами. Одним из таких методов яв-ляется стыковка обычных МДП полевых транзисторов с фрагмен-тами пурпурных мембран, которые представляют собой белковые молекулы бактериородопсина, упакованные в двухслойную ли-пидную мембрану. Основным свойством молекулы бактериоро-допсина 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 поле-вого транзистора, в котором на месте затвора расположена пур-пурная мембрана.

221

Поскольку линейные размеры пурпурных мембран сравнимы с размерами транзисторов (до 1,0 мкм), то такая стыков-ка оказывается вполне реальной. Слои бактериородопсина, со-стоящие 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 электриче-ские.

Основной особенностью перечисленных выше молекуляр-ных носителей, используемой разработчиками запоминающих структур, является наличие двух состояний молекулярной струк-туры. Указанная способность позволяет определять текущее со-стояние молекулы с помощью лазера, настроенного на соответст-вующую частоту.

Рассмотрим систему памяти, в которой молекулярный но-ситель запоминает данные в трехмерной матрице в виде прозрач-ной капсулы, заполненной полиакридным гелем . Один из вариан-тов такой капсулы может иметь размеры 20 х 20 х 40 мм. Для функционирования памяти капсулу помещают в запоминающую структуру, состоящую из нескольких лазеров и детекторной мат-рицы, реализованной на базе прибора, использующего принцип зарядовой инжекции, и позволяющей производить запись и чтение данных.

Перед записью данных с помощью желтого страничного лазера на 590 нм производится активизация возбужденной плоско-сти в материале внутри капсулы. Получение энергоактивной плос-кости в виде страницы данных, требует наличия пространственно-го светового модулятора. Таким модулятором может быть жид-кокристаллическая матрица, создающая маску на пути лазерного луча. Таким образом, при активизации записывающей плоскости происходит перевод молекул в О - состояние, оответствующее значению бита "0".

Для записи информации зажигается красный лазер, перед которым также устанавливается пространственный световой мо-дулятор , отображающий двоичные данные. При этом последний создает на пути луча необходимую маску, способствующую облу-чению только определенных точек страницы. Молекулы в таких

222

местах будут соответствовать двоичной единице в отличие от оставшейся части страницы, представляющей двоичные нули.

Для чтения информации необходимо включить считываю-щий лазер, при этом, из-за различия в спектрах поглощения, мож-но идентифицировать двоичные нули и единицы. Молекулы, представляющие двоичный нуль, 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 си-него лазера, чтобы вернуть молекулы из состояния двоичной еди-ницы в двоичный нуль. Страница данных может быть прочитана без разрушения до 5000 раз, после чего освежается с помощью но-вой записи.

С помощью предложенной запоминающей структуры мож-но получить доступ к данным со скоростью до 90 МВ/с при усло-вии об]единения по восемь запоминающих битовых ячеек в байт с параллельным доступом и соответствующей схемотехнической реализации системы памяти. Емкость данных в рассмотренной кювете может достигать до 1,4 GB, поскольку дальнейшее увели-чение 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 достига-ется существенного увеличения уровня сигнала считывания ин-формации. Сигнал с фоторегистрирующей матрицы обрабатывает-ся и регистрируется на ЭВМ.

Проведенные исследования позволяют говорить о хранении и обработке информации на молекулярном уровне. Идентифика-ция двоичных нулей и единиц производится за счет разности спек-тров поглощений лазерного из- лучения различными состояниями исследуемой молекулярной структуры. При этом молекулы, по-глощающие свет, отвечают двоичному нулю, а молекулы, пропус-кающие лазерный луч мимо себя, отвечают двоичной единице.

Исследуемые молекулярные запоминающие структуры,

223

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 вычисли- тельных систем, могут быть эффективно использованы в различных отрас-лях науки и техники.

Значительные перспективы, которые открывает технология записи и считывания информации на светочувствительном мате-риале, 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 со-держит движущихся частей и доступ к страницам осуществляется параллельно, то запоминающее устройство в целом может позво-лить 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 объ-емнее. И главное, что подобные технологии будут крайне дешевы, в десятки раз дешевле сегодняшних технологий.

Сама идея этих логических элементов не является револю-ционной: кремниевые микросхемы содержат миллиарды точно та-ких же. Но преимущества в потребляемой энергии и размерах спо-собны сделать компьютеры вездесущими. Молекулярный компью-тер размером с песчинку может содержать миллиарды молекул. А если научиться делать компьютеры не трехслойными, а трехмер-ными, преодолев ограничения процесса плоской литографии, при-меняемого для изготовления микропроцессоров сегодня, преиму-щества станут еще больше.

Кроме того, молекулярные технологии сулят появление микромашин, способных перемещаться и прилагать усилие. При-чем для создания таких устройств можно применять даже тради-ционные технологии травления. Когда-нибудь эти микромашины

224

будут самостоятельно заниматься сборкой компонентов моле-кулярного или атомного размера.

Первые опыты с молекулярными устройствами еще не га-рантируют появления таких компьютеров, однако это именно тот путь, который предначертан всей историей предыдущих достиже-ний. Массовое производство действующего молекулярного ком-пьютера вполне может начаться где-нибудь между 2005 и 2015 го-дами.

**8.6.9. Оптические МПС**

Оптические системы в качестве носителя информации ис-пользуют свет. Простейшим способом передачи данных при по-мощи луча света является его модуляция по принципу "есть свет - нет света". У света есть одна особенность: распространяясь в про-странстве, он всегда заполняет собой какую-то его часть. Следова-тельно, существует возможность кодирования информации путем изменения амплитуды и фазы луча не только во времени, но и в пространстве.

Формой реализации этой идеи является оптический эле-мент, называемый транспарантом. Попросту говоря, это кусочек прозрачного материала, на который каким-либо способом нанесе-но изображение, представляющее собой пространственное (в ко-ординатах, связанных с транспарантом) распределение коэффици-ента поглощения, коэффициента преломления (или толщины) или же того и другого одновременно. В первом случае получим транс-парант, модулирующий пространственное распределение ампли-туды проходящего через него света. Примером такого транспаран-та может являться кадрик обычной фотопленки.

Во втором - получаем так называемый фазовый транспа-рант. Как правило, он совершенно прозрачен, и невооруженным глазом невозможно разглядеть на нем никакого изображения. Ма-териал 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

225

возникновению за пластинкой транспаранта заданного про-странственного распределения фаз. Еще говорят о формировании определенного фазового профиля волнового фронта (рис. 54).

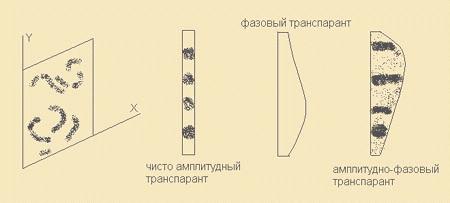


рис. 54

Из сказанного выше становится очевидным, что любой са-мый привычный оптический элемент - будь то линза или целый объектив - представляет собой амплитудно-фазовый транспарант (иногда используют термин "фильтр").

Транспаранты можно изготавливать на базе технологий, по-зволяющих управлять оптическими свойствами вещества (жидкие кристаллы или электрооптические материалы), что дает возмож-ность 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 све-та, мы, по существу, уже касаемся вопросов не только кодирова-ния или представления информации, но и ее обработки.

Само понятие "модуляция" ничего не говорит о том, что происходит с модулированным сигналом дальше. Привычно ду-мать , что ничего особенного с ним не происходит. Распространя-ется себе и все.

226

В данном случае в пространстве за транспарантом свето - вая волна не "просто распространяется". Там, в этой области, ин-терферируют части волны, прошедшие различные участки транс-паранта (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 моду-ляции , сколько о преобразовании структуры волны, то есть, по су-ти, об обработке информации, записанной в этой структуре.

Сейчас мы затронули чрезвычайно важный , я бы сказал, принципиально важный момент. Из математики мы знаем, что лю-бые процессы или объекты могут быть представлены как сумма (ряд) некоторых элементарных периодических функций, напри-мер , гармонических колебаний. Этот ряд называют еще спектром или спектральным представлением объекта.

Любые изменения некоторого объекта, допустим, оптиче-ского изображения, связаны с изменениями его спектрального со-става. Поэтому транспаранты в технике оптических вычислений называют еще пространственными амплитудно-фазовыми фильт-рами, подчеркивая, что они осуществляют некоторый процесс фильтрации, то есть изменения состава пространственных частот исходного изображения. Именно поэтому и говорят, что комбина-ция пространственного фильтра и некоторого объема пространства за ним - уже представляет собой оптический процессор.

Как это все эти принципы работают "на практике". Созда-дим экспериментальную установку: ширма с тремя источниками света А, В и С, экран (см. рис. 55).

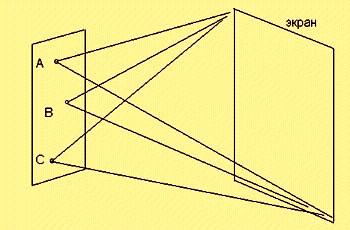


рис. 55

227

Как сделать, чтобы на экран попадал свет только от источ-ников А и С?

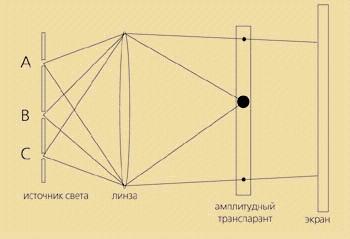


рис. 56

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 установ-ку амплитудным транспарантом-фильтром (попросту говоря, стек-лянной пластинкой с непрозрачным участком напротив изображе-ния 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 конфи-гурации, 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 сфото-графированы все три объекта.

Можно ли сделать так, чтобы фотоаппарат, снимая сцену, не зафиксировал объект В? Как "вычесть" отраженное от него све-товое поле из общего поля сцены? В рамках методов обычной оп-тики эта задача неразрешима.

Для решения этой задачи применяют специальный вид ам-плитудно-фазовых транспарантов, которые называются комплекс-но-сопряженными фильтрами. Математические подробности авто-

228

ры не приводят, но отмечают, что эти оптические элементы об-ладают свойством собирать в одну точку световое поле, обладаю-щее некоторой заранее заданной структурой. Если нам известна конфигурация (амплитудно-фазовое распределение) поля, рассе-янного объектом В, то, изготовив соответствующий фильтр, мож-но решить поставленную задачу, например, так (рис. 57):

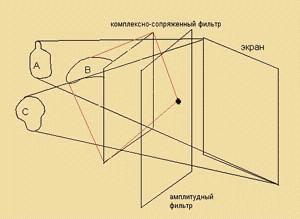


Рис.57

Свет, рассеянный объектом В, собирается в точку, где пере-крывается амплитудным фильтром. Теперь на пленке в фотоаппа-рате будет формироваться все та же сцена, что и раньше, но объек-та 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 го-лографических транспарантах.

Теперь самое время рассмотреть случаи, когда требуется не просто выделение или распознавание требуемого информационно-го 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.

229

Тогда становится очевидным, что если обычная голо-грамма, освещаемая опорным лучом, восстанавливает изображе-ние объекта съемки, то голограмма, полученная путем записи ин-терференции света от двух объектов, будет в свете одного из них восстанавливать изображение другого и наоборот (это свойство носит название "принцип обратимости голограмм").

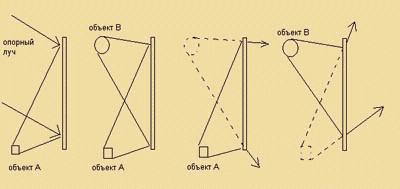


рис. 58

Практическая польза здесь очевидна: получается возмож-ность осуществить " преобразование" любого изображения в зара-нее 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 процес-сора. Массовое производство такого рода устройств по своему ре-волюционному воздействию на отрасль сравнимо с началом вы-пуска интегральных микросхем.

Обработка информации в оптической системе может осу-ществляться как в процессе переноса изображения (представляю-щего собой специальным образом подготовленный входной сиг-

230

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 технологиче-ские нормы) можно получить пиковую производительность по-рядка 1016 элементарных операций в секунду!

Коммутация информационных каналов в оптическом ком-пьютере осуществляется с большой скоростью и отличается про-стотой реализации за счет того, что лучи света в пустом простран-стве 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 чи-пах создает дополнительные возможности, которым у электрон-ных соединений нет никаких аналогов.

Еще одно уникальное свойство оптических систем : в про-зрачной среде информация, закодированная оптическим лучом, может обрабатываться без затрат энергии. Естественно, закон со-хранения энергии при этом не нарушается. Наконец, отметим, что оптическая система ничего не излучает во внешнюю среду, обес-печивая защиту компьютера от перехвата информации . И обратно: оптическая система надежно защищена от сторонних электромаг-нитных наводок.

Основные элементы оптических компьютеров с переносом изображения давно известны. Это - линза, зеркало, оптический транспарант и слой пространства. В настоящее время к ним доба-вились волноводные элементы, а также лазеры, полупроводнико-вые многоэлементные фотоприемники, нелинейные оптические среды, разного рода дефлекторы и светоклапанные устройства.

Базисная логическая функция, элементарный кирпичик, с помощью которого можно построить любой, сколь угодно слож-ный цифровой компьютер, имеет множество оптических реализа-ций. На рис. 59 дан простой пример построения многовходовой функции ИЛИ- НЕ/И-НЕ с помощью линзы L и порогового устрой-ства-инвертора N.

231

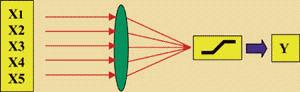


Рис. 59

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 светоизлучаю-щих диодов. Они расположены на фокальной линии цилиндриче-ской линзы L1. T - оптический транспарант с записанной на нем матрицей пропускания T(i, j).

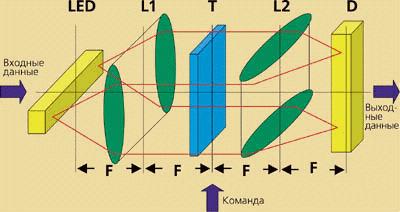


Рис. 60

Строки матрицы параллельны образующей первой линзы. L2 - цилиндрическая линза, образующая которой параллельна столбцам матрицы транспаранта. Она собирает лучи, прошедшие

232

через элементы одной строки, на одном пикселе многоэлемент-ного линейного фотоприемника D. Нетрудно видеть, что входной X и выходной Y вектора связаны линейным преобразованием

Y=TX

* оптической системе возможна также обработка двумер-ных структур. На рис. 61 представлена схема оптического процес-сора, 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 обраба-тываемым изображениям. Распределение интенсивности излуче-ния на матричном фотоприемнике пропорционально интегралу

J(x, y)=тт(T1(x-u, y-v) T2(u, v) du dv.

* предыдущих примерах свет выполнял ту же роль, что и электроны в проводниках обычных микросхем. При этом в качест-ве "проводов" выступали геометрические лучи. Понятно, что с та-ким же успехом свет можно загнать в волновод и организовать вычислительную среду по принципам, близким к идеологии элек-тронной полупроводниковой микросхемотехники. Этим занимает-ся интегральная и волоконная оптика.

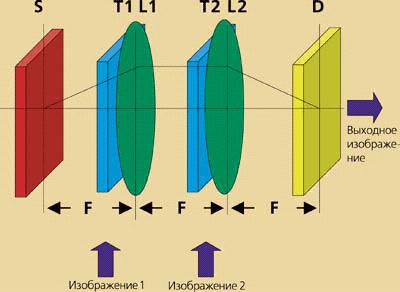


Рис. 61

233

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 ампли-туду проходящей световой волны. L1 и L2 - сферические линзы с фокусным расстоянием F. Результирующий сигнал считывается матричным фотоприемником D.

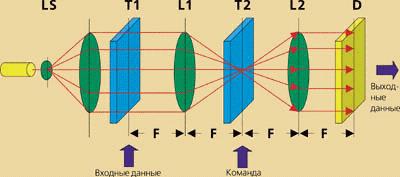


Рис . 62 Распределение амплитуды светового поля в плоскости фо-

тодетектора 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 распознава-ния образов.

Если же нас интересует Фурье-спектр двумерного сигнала, то он вычисляется с помощью линзы L и слоя пространства дли-ной F так, как показано на рис. 63. Остальные элементы предна-значены 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 про-порционально n log(n). В оптическом компьютере эта процедура даже в двумерном случае выполняется всего за один машинный

234

такт, что делает оптический компьютер незаменимым для воен-ных целей, а также для решения задач, требующих быстрой оцен-ки ситуации и управления в реальном времени.

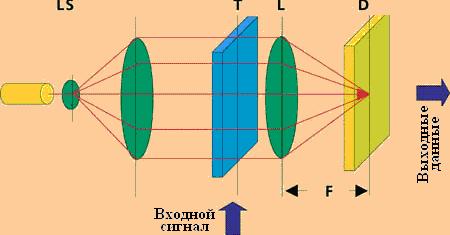


Рис. 63

Замечательным свойством оптического компьютера с пере-дачей изображения является его способность за один такт обраба-тывать двумерные картинки, причем машинная команда сама мо-жет представляться картинкой. Откуда же эти картинки брать? Их источником могут быть внешняя среда, результат предыдущего вычисления, оптическая память. Если из внешней среды изобра-жение можно ввести с помощью объектива, а результат предыду-щего вычисления возвратить на вход процессора с помощью сис-темы зеркал, то в качестве памяти можно использовать различные оптические и оптоэлектронные устройства записи, хранения и из-влечения изображений. Например, кассету со слайдами.

Особый интерес представляют голографические устройства памяти. Такая память обладает рядом достоинств. Голограмма со-храняет информацию не только об интенсивности, но и о фазе све-товой волны, что в оптике принципиально важно, а с утилитарной точки зрения - позволяет повысить объем записываемой информа-ции. Кроме того, различные картинки можно записывать в одно и то же место, используя весь объем носителя, а не тонкий слой по-

235

верхности (как в случае обычной оптической или магнитной памяти). По оценкам специалистов, объемная плотность записи информации может превышать величину 1011 бит/см3, а скорость ввода информации с голограмм - несколько гигабит в секунду.

Кроме того, голограмма сама может использоваться в каче-стве принципиального узла оптического процессора, выступая од-новременно в роли буферной памяти и обрабатывающего элемен-та. Информация в таких голограммах записывается путем измене-ния показателя преломления по всему объему носителя. Обычно для этого используются прозрачные материалы с ярко выражен-ными нелинейными оптическими характеристиками, например, кристаллы ниобата лития.

На рис. 64 показано примерное устройство оптической па-мяти с объемной голографической средой. Ввод информации осу-ществляется с помощью управляемого оптического транспаранта. Адресацией при записи-считывании управляет опорный луч. Счи-тываемая информация фокусируется в плоскости многоэлементно-го матричного фотоприемника D.

* + силу того, что емкость голографической памяти огромна,
* время выборки мало, кардинально меняется весь подход к орга-низации 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 парал-лельную обработку сложных структур данных.

Речь пойдет об интерфейсах. Может показаться, что опти-ческая вычислительная среда, обладая уникальной способностью одновременно обрабатывать большие массивы информации, как бы оторвана от хорошо разработанной и всенародно любимой электронной вычислительной инфраструктуры, в которой обра-ботка и передача информации осуществляется последовательно. Однако на самом деле любой оптический компьютер должен быть обрамлен обычной электроникой.

236

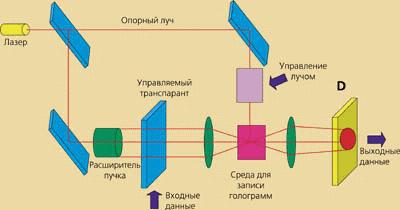


Рис. 64

Сегодня при вводе оптического изображения в ЭВМ ваку-умные передающие телевизионные трубки полностью вытеснены приборами с зарядовой связью ПЗС (CCD - charge coupled device). Устройства на основе ПЗС воспринимают изображение парал-лельно по оптическому каналу, а передают последовательно по электронному. В ряде случаев последовательная перекачка содер-жимого ПЗС в оперативную память ЭВМ ограничивает быстро-действие всего оптоэлектронного вычислительного комплекса. В настоящее время имеются российские и зарубежные разработки оптоэлектронных интерфейсов, в которых матричный фотоприем-ник конструктивно совмещен с элементами RAM в оперативной памяти ЭВМ. Отсюда - один шаг до использования оптического канала доступа к электронным чипам через третье измерение. Та-кие работы сегодня ведутся в зарубежных лабораториях. Оказыва-ется, даже в стандартной персоналке длина соединений между вентилями микросхем , расположенных на разных печатных пла-тах, порой измеряется метрами. Использование оптического кана-ла для организации "междучипового" обмена позволяет на порядок сократить задержку распространения сигнала.

От ЭВМ к оптической системе информация передается с помощью управляемых транспарантов, или пространственных мо-

дуляторов света ПМС (SLM - spatial light modulator). Однако жид-

кокристаллические панели, чаще всего использующиеся для этих целей, обладают существенным недостатком - сравнительно низ-ким быстродействием. Существуют также оптические транспаран-

237

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 помо-щью второй световой волны. По сути - это прозрачная интеграль-ная схема, выполненная по оптоэлектронной технологии и исполь-зующая параллельный оптический вход и выход информации с помощью третьего измерения. Интересно, что для первых экспе-риментов с устройствами типа SLM-SP обычные чипы кремниевой памяти покрывали тонким слоем жидкого кристалла. Электриче-ское поле вблизи заряженных ячеек памяти ориентирует молекулы жидкого кристалла, обнаруживая электрооптический эффект в от-раженном 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 много-элементные лазеры. Последнее достижение в этой области - мат-рицы полупроводниковых лазеров размером 1000х1000 светоизлу-чающих элементов. Излучение генерируется перпендикулярно плоскости чипа. В таких оптоэлектронных БИС используется тех-нология изготовления лазеров с вертикальным расположением ре-

зонаторов (VCSEL, vertical cavity surface-emitting lasers). Недоста-

ток подобных устройств состоит в том, что в каждый момент вре-мени 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 некото-рых наиболее эффектных своих применениях оптический процес-сор является аналоговой машиной. С другой стороны, способность оптической вычислительной среды образовывать параллельные связи между большим числом элементов и одновременно выпол-нять операции типа взвешивания и суммирования аналоговых сиг-налов является идеальной для построения нейросетей. Если в сис-

238

тему ввести транспарант, пропускание которого задано значе-ниями весовых коэффициентов W(i), получим простейшую модель нейрона - персептрон, показанный на рис. 65.

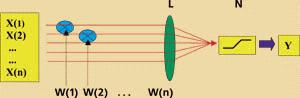


Рис. 65

Аналогично схему, представленную на рис. 65, можно рас-сматривать как слой нейронов. Для этого элементы фотоприемни-ка 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, при-сущая аналоговым компьютерам, не имеет большого значения.

Объединяя рассмотренные оптические схемы в многослой-ные структуры и вводя в них обратные связи, можно создавать сложные быстродействующие нейросети. А на их основе - нейро-компьютеры широкого применения. Это направление считается одним из самых перспективных в области оптического компьюте-ростроения.

Успехи оптоэлектроники последнего десятилетия позволи-ли 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 про-изводства оптических компьютеров для широкого использования не наблюдается. Причина первая: долгое время работы в этой об-

239

ласти были закрытыми. Вторая причина проистекает из первой: никто не занимался маркетинговой проработкой оптического ком-пьютера для народа.

Стоимость оптического компьютера довольно велика. Главные области его использования - военная техника, кримино-логия, защита информации, банковская сфера, а также научные ис-следования. Основные работы за рубежом ведутся в США, Япо-нии, Западной Европе. В США одним из ведущих координаторов работ является известное Агентство по передовым оборонным ис-

следовательским 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 . Ре-шаемые задачи: машинное зрение, искусственный интеллект, рас-познавание 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 голо-граммы.

Сотрудники Коннектикутского университета и Оптического научного центра Аризонского университета предложили преобра-зовывать изображение отпечатков пальцев, лица, подписи вла-дельца документа в специальный фазовый профиль, который од-новременно 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 невоз-можно.

Уже в 1996 году фирмой IBM была разработана голографи-ческая память. Для ввода картинок в память использовался управ-ляемый транспарант размером 1024х1024 пикселей, способный ра-ботать с частотой 1 тыс . изображений в секунду. Для вывода ин-формации применялась камера ПЗС размером 2048х2048 пиксе-лей.

240

Фирмой OptiComp (США) разработан оптоэлектронный 32-битный RISC-процессор, способный обеспечить независимые соединения между 8192 оптическими каналами. Его производи-тельность составляет около 1012 двоичных операций в секунду. Ра-ботая 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 ис-следовательского института JILA (США) оптический нейроком-пьютер используется для распознавания человеческой речи . Фу-рье-спектры отрезков человеческой речи преобразовывались в ви-зуальную картину, которая далее обрабатывалась оптическими ме-тодами. Система работала гораздо быстрее и надежнее, чем специ-ально обученная нейросеть на электронных чипах.

Несмотря на определенные сложности в разработке и изго-товлении оптических компьютерных систем, идея его создания не умерла. Более того, сейчас она снова стала актуальной. Дело в том, что, увеличивая и увеличивая быстродействие процессора, произ-водители с той же неизбежностью вынуждены работать со все бо-лее и более мелкими объектами. Сейчас отметка находится на ру-беже 0,06 мкм , но это, разумеется, не предел. Нанотехнология, ко-торая вначале тоже рассматривалась как некая полуфантастиче-ская технология, постепенно превратилась в обычную задачу для современных технологов. Вот тут и выяснилось, что в нанотехно-логии оптические подходы очень даже хороши.

Это можно объяснить следующим. С быстродействием у оптических компонентов все отлично. Если тактовая частота со-временных компьютерных систем только сейчас подошла к 3-4 ГГц, то оптические переключатели преодолели этот рубеж еще лет двадцать назад. Эффективность работы оптических переключате-лей можно видеть на примере волоконно-оптических линий связи. Здесь оптическая технология окончательно победила технологию электрическую.

Наибольшее распространение получили переключатели на основе LiNbO3. Ниобат лития - уникальный кристалл. У него от-личные электрооптические характеристики, он обладает замеча-тельными акустооптическими свойствами и вдобавок ко всему еще

241

и нелинейными характеристиками. Поэтому в системах обра - ботки оптической информации устройства на основе ниобата ли-тия встречаются чаще всего.

* области создания элементов памяти для оптических сис-тем тоже достигнуты значительные успехи, более того, в области оптической памяти в последние годы получены особенно интерес-ные результаты. Активизация этих работ была стимулирована раз-витием носителей информации, а именно лазерных компакт-дисков и систем магнитооптической записи. Поскольку CD и маг-нитооптические 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 соот-ветствует 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 Ангст-рем. Счет идет просто на отдельные атомы. Полученные результа-ты действительно впечатляют: если эти разработки дойдут до про-мышленного производства, то 3,5-дюймовые диски емкостью в де-сятки гигабайт станут вполне доступны.

Есть еще одно направление оптической памяти - гологра-фическая 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 сей-час создают целые системы голографической памяти. Число голо-грамм на одном кристалле достигло десяти тысяч, и это не предел.

* оптических системах проблема интеграции является наи-более важной и в конечном итоге решающей. Когда все компонен-ты разработаны и испытаны, возникает очередная проблема - со-брать все воедино. Для оптического компьютера эта проблема зна-

242

чительно сложнее, чем для обычного. Электрические связи реа-лизовать гораздо проще. Можно, конечно, все элементы объеди-нить с помощью оптического волокна (и такие технологии тоже развиваются), но в таких системах существуют довольно значи-тельные 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 произ-водственные мощности. Нужна лишь база для производства. По-этому любые новые оптические разработки на основе кремния принимаются обычно на ура. Но у кремния есть один хронический недостаток: он плохо совместим с активными оптическими эле-ментами. Оптические усилители и переключатели очень плохо уживаются с кремниевой технологией.

Следующее направление - создание базы на основе альтер-нативных полупроводниковых структур. Здесь есть свои очень важные плюсы: можно формировать структуры с совершенно раз-ными свойствами, встраивать их в оптические интегральные схе-мы (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 пе-реориентации разработчиков систем на кремниевой основе на со-вершенно новое направление. Вторая - экономическая. Главной причиной убедительной победы кремниевой технологии была де-шевизна. 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 бу-дущем оптический компьютер, или нет. У оптических компьюте-ров 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

243

информации главной проблемой является трафик. Число поль-зователей 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 пере-дачи и обработки информации. Пропускная же способность опти-ческих систем растет, и возможности роста колоссальны.

Один из основных способов увеличения пропускной способ-ности - системы спектрального уплотнения. И не зря было упомя-нуто, что на структурах из ниобата лития делают не только пере-ключатели, 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 архитек-турой MSIMD, в которой реализованы два принципиальных тех-нических решения: упрощен до уровня нейрона процессорный элемент однородной структуры и резко усложнены связи между элементами; программирование вычислительной структуры пере-несено на изменение весовых связей между процессорными эле-ментами.

Общее определение нейрокомпьютера может быть пред-ставлено в следующем виде . Нейрокомпьютер - это вычислитель-ная система с архитектурой аппаратного и программного обеспе-чения, адекватной выполнению алгоритмов, представленных в нейросетевом логическом базисе

Каждый нейрон получает сигналы от соседних нейронов по специальным нервным волокнам. Эти сигналы могут быть возбу-ждающими или тормозящими. Их сумма составляет электрический потенциал внутри тела нейрона. Когда потенциал превышает не-который порог, нейрон переходит в возбужденное состояние и по-сылает сигнал по выходному нервному волокну. Отдельные искус-

244

ственные нейроны соединяются друг с другом различными ме-тодами. Это позволяет создавать разнообразные нейронные сети с различной архитектурой, правилами обучения и возможностями.

Термин “искусственные нейронные сети” у многих ассо-циируется с фантазиями об андроидах и бунте роботов, о маши-нах, заменяющих и имитирующих человека. Это впечатление уси-ливают 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 эле-ментов формальных нейронов. Большая часть работ по нейроин-форматике 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 слож-ность мозга, гибкость его функционирования и другие важнейшие качества определяются связями между нейронами. Каждая связь представляется как совсем простой элемент, служащий для пере-дачи сигнала. Коротко эту мысль можно выразить так: “структура связей все, свойства элементов ничто”.

Совокупность идей и научно-техническое направление, оп-ределяемое описанным представлением о мозге, называется кон-некционизмом (connection связь). С реальным мозгом все это со-относится примерно так же, как карикатура или шарж со своим прототипом. Важно не буквальное соответствие оригиналу, а про-дуктивность технической идеи.

С коннекционизмом тесно связан следующий блок идей:

* однородность системы (элементы одинаковы и чрезвы-чайно просты, все определяется структурой связей);
* надежные системы из ненадежных элементов и “аналого-вый 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.

245

Для описания алгоритмов и устройств в нейроинформа-тике выработана специальная “ схемотехника”, в которой элемен-тарные устройства (сумматоры, синапсы, нейроны и т.п.) объеди-няются в сети, предназначенные для решения задач. Для многих начинающих кажется неожиданным, что ни в аппаратной реализа-ции нейронных сетей, ни в профессиональном программном обес-печении эти элементы вовсе не обязательно реализуются как от-дельные части или блоки. Используемая в нейроинформатике иде-альная 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.

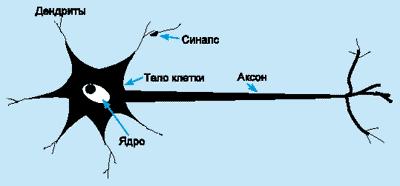


Рис. 66 Нейрон (нервная клетка) является особой биологической

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 молекулярны-ми средствами для производства необходимых нейрону материа-лов. Нейрон получает сигналы (импульсы) от других нейронов че-рез дендриты (приемники) и передает сигналы, сгенерированные телом клетки , вдоль аксона (передатчик), который в конце раз-ветвляется на волокна (strands). На окончаниях этих волокон нахо-дятся синапсы (synapses).

246

Синапс является элементарной структурой и функцио-нальным узлом между двумя нейронами (волокно аксона одного нейрона и дендрит другого). Когда импульс достигает синаптиче-ского окончания, высвобождаются определенные химические ве-щества, называемые нейротрансмиттерами . Нейротрансмиттеры диффундируют через синаптическую щель, возбуждая или затор-маживая, в зависимости от типа синапса, способность нейрона-приемника генерировать электрические импульсы. Результатив-ность синапса может настраиваться проходящими через него сиг-налами, так что синапсы могут обучаться в зависимости от актив-ности процессов, в которых они участвуют. Эта зависимость от предыстории действует как память, которая, возможно, ответст-венна за память человека.

Кора головного мозга человека является протяженной, об-разованной нейронами поверхностью толщиной от 2 до 3 мм с площадью около 2200 см2, что вдвое превышает площадь поверх-ности стандартной клавиатуры. Кора головного мозга содержит около 1011 нейронов, что приблизительно равно числу звезд Млечного пути. Каждый нейрон связан с 103 - 104 другими нейро-нами . 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 моду-ляции. Частота может изменяться от нескольких единиц до сотен герц , что в миллион раз медленнее, чем самые быстродействую-щие переключательные электронные схемы. Тем не менее , слож-ные решения по восприятию информации, как, например , распо-знавание лица, человек принимает за несколько сотен мс. Эти ре-шения контролируются сетью нейронов, которые имеют скорость выполнения операций всего несколько мс. Это означает, что вы-числения требуют не более 100 последовательных стадий. Други-ми словами, для таких сложных задач мозг "запускает" параллель-ные программы, содержащие около 100 шагов. Это известно как правило ста шагов. Рассуждая аналогичным образом, можно обна-ружить, что количество информации, посылаемое от одного ней-рона другому, должно быть очень маленьким (несколько бит). От-

247

сюда следует, что основная информация не передается непо-средственно, 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), кото-рый, впоследствии, и лег в основу нейрокомпьютера.

Искусственный нейрон имитирует в первом приближении свойства биологического нейрона. На вход искусственного нейро-на 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 нейро-на.

На рис. 67 представлена модель реализующая эту идею. Хо-тя сетевые парадигмы весьма разнообразны, в основе почти всех их лежит эта конфигурация . Здесь множество входных сигналов, обозначенных x1, x2, x3...xn, поступает на искусственный нейрон. Эти входные сигналы, в совокупности обозначаемые вектором **X**, соответствуют сигналам, приходящим в синапсы биологического нейрона. Каждый сигнал умножается на соответствующий вес w1, w2, w3...wn, и поступает на суммирующий блок (**Σ) (**адаптивный сумматор**)**. Каждый вес соответствует "силе" одной биологической синаптической связи (множество весов в совокупности обознача-ется вектором **W**).

Суммирующий блок, соответствующий телу биологическо-го элемента, складывает взвешенные входы алгебраически, созда-вая на выходе сигнал **XW**. Он преобразуется активационной функцией **F** и дает выходной нейронный сигнал **OUT** .

Активационная функция может быть обычной линейной функцией:

**OUT=K(XW),**

где **К** - постоянная, пороговой функцией,

**OUT=1**,если **XW>T, OUT=0** в остальных случаях,

248

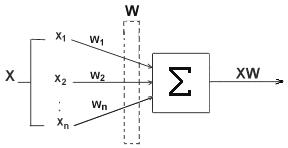


Рис. 67.

где **T** - некоторая постоянная пороговая величина, или же функцией более точно моделирующей нелинейную передаточную характеристику биологического нейрона и представляющей ней-ронной сети большие возможности.

На рис. 68 блок обозначенный **F** , принимает сигнал **XW** и выдает сигнал **OUT**. Если блок **F** сужает диапазон изменения ве-личины **XW** так, что при любых значениях **XW** значения **OUT** принадлежат некоторому конечному интервалу, то **F** называется сжимающей функцией.

В качестве сжимающей функции часто используется логи-стическая 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).**

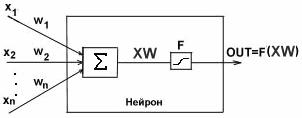


Рис. 68

249

По аналогии с электронными системами активационную функцию можно считать нелинейной усилительной характеристи-кой искусственного нейрона. Коэффициент усиления вычисляется как отношение приращения величины OUT к вызвавшему его не-большому приращению величины XW. Он выражается наклоном кривой при определенном уровне возбуждения и изменяется от малых значений при больших отрицательных возбуждениях (кри-вая почти горизонтальна) до максимального значения при нулевом возбуждении и снова уменьшается, когда возбуждение становится большим положительным.

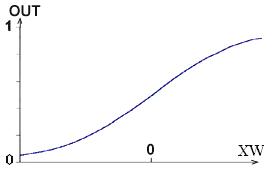


Рис. 69.

Гроссберг (1973) обнаружил, что подобная нелинейная ха-рактеристика решает поставленную им дилемму шумового насы-щения . Слабые сигналы нуждаются в большом сетевом усилении, чтобы дать пригодный к использованию выходной сигнал. Однако усилительные каскады с большими коэффициентами усиления мо-гут привести к насыщению выхода шумами усилителей (случай-ными флуктуациями), которые присутствуют в любой физически реализованной сети. Сильные входные сигналы в свою очередь также будут приводить к насыщению усилительных каскадов, ис-ключая возможность полезного использования выхода. Централь-ная область логистической функции, имеющая большой коэффи-циент усиления, решает проблему обработки слабых сигналов, в то время как в области с падающим усилением на положительном и

250

отрицательном концах подходят для больших возбуждений. Та-ким образом, нейрон функционирует с большим усилением в ши-роком диапазоне уровня входного сигнала.

Рассмотренная простая модель искусственного нейрона иг-норирует многие свойства своего биологического двойника. На-пример, она не принимает во внимание задержки во времени , ко-торые воздействуют на динамику системы. Входные сигналы сра-зу же порождают выходной сигнал. И что более важно, она не учитывает воздействий функции частотной модуляции или син-хронизирующей функции биологического нейрона , которые ряд исследователей считают решающими. Несмотря на эти ограниче-ния, 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 возбу-ждающими или тормозящими. Их сумма составляет электрический потенциал внутри тела нейрона. Когда потенциал превышает не-который порог, нейрон переходит в возбужденное состояние и по-сылает сигнал по выходному нервному волокну . Отдельные искус-ственные нейроны соединяются друг с другом различными мето-дами. Это позволяет создавать разнообразные нейронные сети с различной архитектурой, правилами обучения и возможностями.

Искусственная Нейронная Сеть (ИНС) может рассматри-ваться как направленный граф со взвешенными связями, в котором искусственные нейроны являются узлами. По архитектуре связей ИНС могут быть сгруппированы в два класса: сети прямого рас-пространения, 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 рас-положены слоями и имеют однонаправленные связи между слоя-ми. На рисунке представлены типовые сети каждого класса. Сети прямого распространения являются статическими в том смысле, что на заданный вход они вырабатывают одну совокупность вы-

251

ходных значений, не зависящих от предыдущего состояния се-ти. Рекуррентные сети являются динамическими, так как в силу обратных связей в них модифицируются входы нейронов, что при-водит к изменению состояния сети.

Здесь каждый нейрон передает свой выходной сигнал ос-тальным 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 полно-связных сетей могут выбираться по-разному. Существует, впро-чем, стандартный выбор: нейрон с адаптивным неоднородным ли-нейным сумматором на входе. Для полносвязной сети входной сумматор нейрона фактически распадается на два: первый вычис-ляет линейную функцию от входных сигналов сети, второй линей-ную функцию от выходных сигналов других нейронов, получен-ных на предыдущем шаге . Функция активации нейронов (характе-ристическая функция) это нелинейный преобразователь выходного сигнала сумматора. Если функция одна для всех нейронов сети, то сеть называют однородной (гомогенной). Если же характеристиче-ская функция зависит еще от одного или нескольких параметров, значения которых меняются от нейрона к нейрону, то сеть назы-вают неоднородной (гетерогенной).

Составлять сеть из нейронов стандартного вида не обяза-тельно. Слоистая или полносвязная архитектуры не налагают су-щественных ограничений на участвующие в них элементы. Един-ственное жесткое требование, предъявляемое архитектурой к эле-ментам сети, это соответствие размерности вектора входных сиг-налов элемента (она определяется архитектурой) числу его входов. Если полносвязная сеть функционирует до получения ответа за-данное число тактов k, то ее можно представить как частный слу-чай 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 функциониро-вания заранее не ограничено и слоистая сеть так работать не мо-жет. Доказаны теоремы о полноте: для любой непрерывной функ-ции нескольких переменных можно построить нейронную сеть,

252

которая вычисляет эту функцию с любой заданной точностью. Так что нейронные сети в каком-то смысле могут все.

Способность к обучению является фундаментальным свой-ством мозга. В контексте ИНС процесс обучения может рассмат-риваться как настройка архитектуры сети и весов связей для эф-фективного выполнения специальной задачи. Обычно нейронная сеть должна настроить веса связей по имеющейся обучающей вы-борке. Функционирование сети улучшается по мере итеративной настройки весовых коэффициентов. Свойство сети обучаться на примерах делает их более привлекательными по сравнению с сис-темами, которые следуют определенной системе правил функцио-нирования, сформулированной экспертами.

Для конструирования процесса обучения, прежде всего, не-обходимо иметь модель внешней среды, в которой функционирует нейронная сеть - знать доступную для сети информацию. Эта мо-дель определяет парадигму обучения. Во-вторых, необходимо по-нять, как модифицировать весовые параметры сети - какие прави-ла 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 каж-дый входной пример. Веса настраиваются так, чтобы сеть произ-водила 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 . Обуче-ние без учителя не требует знания правильных ответов на каждый пример обучающей выборки. В этом случае раскрывается внут-ренняя структура данных или корреляции между образцами в сис-теме данных, что позволяет распределить образцы по категориям. При смешанном обучении часть весов определяется посредством обучения с учителем, в то время как остальная получается с по-мощью самообучения.

Теория обучения рассматривает три фундаментальных свойства, связанных с обучением по примерам: емкость, слож-

253

ность образцов и вычислительная сложность. Под емкостью понимается, сколько образцов может запомнить сеть, и какие функции и границы принятия решений могут быть на ней сформи-рованы. Сложность образцов определяет число обучающих при-меров, необходимых для достижения способности сети к обобще-нию. Слишком малое число примеров может вызвать "переобу-ченность" сети, когда она хорошо функционирует на примерах обучающей выборки, но плохо - на тестовых примерах, подчинен-ных 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 уменьше-ние ошибки. Обучение имеет место только в случае, когда перцеп-трон ошибается. Известны различные модификации этого алго-ритма обучения.
* обучение Больцмана представляет собой стохастическое правило обучения, которое следует из информационных теорети-ческих и термодинамических принципов. Целью обучения Больц-мана является такая настройка весовых коэффициентов, при кото-рой состояния видимых нейронов удовлетворяют желаемому рас-пределению вероятностей. Обучение Больцмана может рассматри-ваться 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 ак-тивизируются 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-

254

даться одновременно, при соревновательном обучении выход-ные нейроны соревнуются между собой за активизацию. Это явле-ние известно как правило "победитель берет все ". Подобное обу-чение 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 ко входному примеру.

Одной из особенностей нейросетевых методов обработки информации является высокая параллельность вычислений и, сле-довательно, целесообразность использования специальных средств аппаратной поддержки. В значительной мере успех в решении рассмотренных задач обусловлен использованием оригинальных ускорительных плат. Такие платы работают параллельно с процес-сором обыкновенного ПК и несут на себе основную вычислитель-ную нагрузку, превращая основной процессор компьютера в уст-ройство управления и обслуживания мощных вычислительных средств, расположенных на ускорительной плате.

Например в НТЦ "Модуль" разработаны многопроцессор-ные 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. Каж-дый процессор имеет свою локальную статическую память объе-мом 1 Мбайт. К 2 процессорам дополнительно подключены 2 бло-ка динамической памяти объемом 16 Мбайт каждый. К одному из процессоров подключена также статическая память объемом 1 Мбайт, используемая для обмена данными с ПК. Процессоры со-единены друг с другом специальными высокоскоростными кана-лами с пропускной способностью 20 Мбайт/с каждый. Наращива-ние и комплексирование плат осуществляется на материнской пла-те ПК с помощью шины ISA.

Ускорительная плата МЦ5.002 содержит 6 процессоров TMS320C40 и выполнена в конструктиве VME, что позволяет ис-

255

пользовать ее в бортовых системах, расположенных на лета-тельном аппарате.

Нейропроцессор обычно состоит из двух основных блоков: скалярного, выполняющего роль универсального вычислительного устройства, и векторного, ориентированного на выполнение век-торно-матричных операций. Скалярное устройство обеспечивает интерфейсы с памятью и коммуникационными портами , позво-ляющими объединять процессоры в вычислительные сети различ-ной 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 ра-боты с регистровыми парами.

Центральным звеном нейропроцессора является целочис-ленное векторное устройство, обладающее возможностями обра-ботки данных различной разрядности. Оно оперирует n-разрядными словами. Таким образом, процессор рассчитан на вы-сокопроизводительную обработку больших массивов целочислен-ных данных. К примеру отечественный нейропроцессор, разрабо-танный в НТЦ "Модуль ". Скалярное устройство обеспечивает ин-терфейсы с памятью и 2 коммуникационными портами. Скалярное устройство имеет адресных регистров и такое же количество реги-стров общего назначения разрядностью 32 бита каждый.

Центральным звеном нейропроцессора является целочис-ленное векторное устройство, обладающее возможностями обра-ботки данных различной разрядности . Оно оперирует 64-разрядными словами, которые могут быть разбиты на целочислен-ные составляющие практически произвольной разрядности в пре-делах от 1 до 64 бит. На каждую инструкцию векторного процес-сора затрачивается от 1 до 32 тактов. При этом одновременно об-рабатывается до 32 64-разрядных слов. Для организации непре-рывной подачи данных в операционное устройство (ОУ) векторно-го процессора используются внутренние блоки памяти, называе-мые векторными регистрами . Они выполняют роль буфера опе-рандов, буфера для хранения матрицы весов , очереди результатов. При выполнении команды в операционном устройстве операнды по очереди извлекаются из внутреннего буфера и подаются на

256

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 так-товая частота 33 МГц. На специальных векторно-матричных опе-рациях он дает увеличение производительности в десятки раз по сравнению с процессором TMS320C40. Благодаря наличию ком-муникационных портов с интерфейсом , идентичным портам TMS320C40, нейропроцессор может быть интегрирован в гетеро-генную 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 про-граммирования. Программисту приходится оперировать другими категориями , по-другому строить логику программы, решать зада-чи, которые не могли возникнуть при традиционном программи-ровании. Перед ним стоит задача - максимально эффективно ис-пользовать ресурсы вычислительной системы, правильно распре-делить нагрузку между процессорами, задействовать их специ-фичные возможности.

Здесь на первый план выходят методы параллельной обра-ботки данных. Причем слова "параллельная обработка" можно по-нимать 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 выпол-няемые процессы не блокировали друг друга.

Трудности, возникающие при программировании много-процессорных систем, хорошо известны: синхронизация парал-

257

лельных процессов, механизмы обмена данными, проблемы "критических участков", когда несколько процессов задействуют одни и те же ресурсы. Еще одной важной особенностью современ-ных процессоров является высокая разрядность операндов, напри-мер 64 бита, что позволяет размещать в них по несколько малораз-рядных элементов данных и обрабатывать их параллельно. Приме-ром эффективного использования отмеченной особенности совре-менных процессоров является технология MMX, где 64-разрядный регистр разбивается на 8 независимых байтов или на 4 16-битных слова, которые обрабатываются параллельно. Независимость эле-ментов состоит в том, что при смещениях или вычитании не про-исходит заимствования битов у соседних элементов.

По сравнению с обычными компьютерами, нейрокомпью-теры обладают рядом преимуществ. Во первых — высокое быст-родействие, связанное с тем, что алгоритмы нейроинформатики обладают высокой степенью параллельности. Во вторых — нейро-системы делаются очень устойчивыми к помехам и разрушениям. В третьих — устойчивые и надежные нейросистемы могут созда-ваться из ненадежных элементов, имеющих значительный разброс параметров.

Несмотря на перечисленные выше преимущества, эти уст-ройства имеют ряд недостатков:

1. Они создаются специально для решения конкретных за-дач, 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 со-бой просто персональный компьютер или рабочую станцию, в со-став которых входит дополнительная нейроплата. К их числу от-носятся, например, компьютеры серии FMR фирмы Fujitsu. Такие системы имеют бесспорное право на существование, поскольку их возможностей вполне достаточно для разработки новых алгорит-

258

мов и решения большого числа прикладных задач методами нейроматематики.

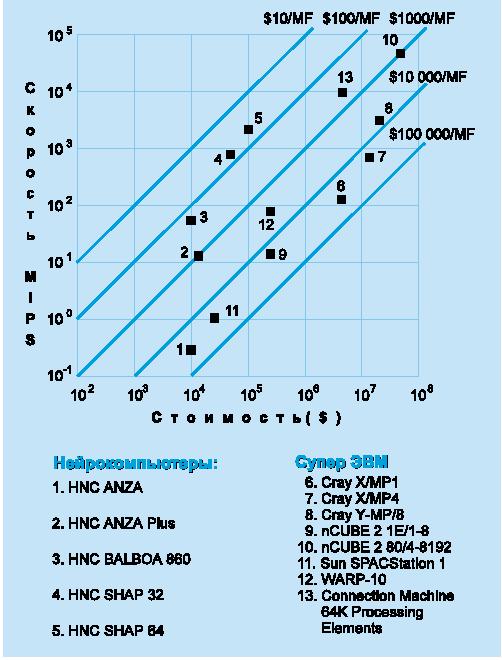


Рис. 70 Однако наибольший интерес представляют специализиро-

ванные нейрокомпьютеры, непосредственно реализующие прин-ципы НС. Типичными представителями таких систем являются

259

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 объе-динены 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 конвей-ерной 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), обеспечи-

вающую 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 моде-лирования таких моделей НС, как сеть Хопфилда - Кохонена и НС с обратным распространением. Его производительность достигает 450 млн. межсоединений/с.

Фирма Computer Recognitiion Systems (CRS) продает серию нейрокомпьютеров WIZARD/CRS 1000, предназначенных для об-работки видеоизображений. Размер входной изображения 512 x 512 пиксел. Модель CRS 1000 уже нашла применение в промыш-ленных 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-

260

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, Т3Е-1350.

Каждый узел в системе содержит один процессорный эле-мент (ПЭ), включающий процессор, память и средство коммута-

ции, которое осуществляет связь между ПЭ. Система конфигури-руется до 2048 процессоров. Пиковая производительность состав-ляет 2,4 Тфлоп.

Разделяемая, высокопроизводительная, глобально адресуе-мая подсистема памяти делает возможным обращение к локальной памяти каждого ПЭ в Cray T3E. Процессорные элементы в системе Cray T3E связаны в трехмерный тор двунаправленной высокоско-ростной сетью с малым временем задержки, которая в шесть раз превосходит по скорости аналогичную сеть в Cray T3D. Также до-бавлена адаптивная маршрутизация, при которой возможен обход участков с высокой эффективностью передачи.

Cray T3E выполняют операции ввода/вывода через много-численные порты на один или более каналы GigaRing. Каналы ввода/вывода интегрированы в 3-х мерную межузловую сеть и пропорциональны размеру системы. При этом при добавлении ПЭ пропускная способность каналов ввода/вывода увеличивается и масштабируемые приложения могут выполнятся на системах с большим числом процессоров также эффективно, как на системах с меньшим числом процессоров.

261

Для Cray T3E была создана масштабируемая версия опе-рационной системы ОС UNICOS — ОС UNICOS/mk. Операцион-ная система UNICOS/mk разделена на программы-серверы, рас-пределенные среди процессоров Cray T3E. Это позволяет управ-лять набором ресурсов системы как единым целым. Локальные серверы обрабатывают запросы ОС, специфичные для каждого ПЭ. Глобальные серверы обеспечивают общесистемные возмож-ности 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 систем-ных ПЭ и не дублируются по всей системе, UNICOS/mk эффек-тивно масштабируема, полно функциональна и обслуживает от де-сятков до тысячи ПЭ с минимальным перегрузкой.

UNICOS/mk обеспечивает следующие программные функ-

ции:

* + Распределение серверов управления файлами. Функции файлового сервера распределены, используя локальные файловые программы-сервера, для обеспечения максимальной производи-тельности и эффективности.
  + ПЭ может генерировать не только последовательную, но
* 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 диско-вые 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 про-цессор 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.

262

Система предназначается для наиболее важных научных

* технических задач в аэрокосмической, автомобильной, финансо-вой, химико-фармацевтической, нефтяной и т.д. промышленно-стях, также в широких областях прикладных исследований, вклю-чая химию, гидродинамику, предсказание погоды и сейсмические процессы.

Для поддержки масштабируемости используется оператив-ная система UNICOS/mk — масштабируемая версия UNICOSR. T3E-1200 поддерживает как явное распараллеливание распреде-лённой памяти посредством CF90 и C/C++ с передачей собщений (MPI, MPI-2 и PVM) и передачу данных, так и неявное распарал-леливание посредством возможностей HPF и Cray CRAFT.

T3E выполняет операции ввода/вывода через многочислен-ные порты на один и более каналов посредством интерфейса GigaRing. Каждый канал сдвоенного кольца ввода/вывода, содер-жащий в двух кольцах данные, которые перемещаются в противо-положных направлениях, передают данные ввода/вывода с высо-кой пропускной способностью и повышенной надёжностью. Все каналы ввода/вывода доступны и управляемы всеми процессор-ными элементами. В Т3Е каждый интерфейс GigaRing имеет мак-симальную пропускную способность в 500 Мбайт/с.

* + дополнение к высокой производительности и пропускной способности процессорных элементов и высокой масштабируемо-сти, Cray T3E-1200 имеет две уникальные особенности: STREAMS
* E-Регистры. STREAMS доводят до максимума пропускную спо-собность локальной памяти, позволяя микропроцессору запускать при полной скорости для ссылки для вектороподобных данных. Е-Регистры предоставляют операции gather/scatter (соедине-ние/вразброс) для ссылок на локальную и удалённую память, и ис-пользуют полную пропускную способность внутреннего соедине-ния для удалённого чтения и записи отдельного слова.

Оценка производительности системы производилась при решении плотной линейной системы уравнений порядка 148800 на машине Т3Е-1200 с 1200 процессорами. Была достигнута скорость в 1,127 Тфлоп/с, что составляет 63% эффективности.

263

* 1. **Многопроцессорные системы**

1. **Общие требования, предъявляемые к МПС**

**Отношение стоимость / производительность.** Появлениелюбого нового направления в вычислительной технике определя-ется требованиями компьютерного рынка. Поэтому у разработчиков компьютеров нет одной единственной цели.



Большая универсальная вычислительная машина (мейнфрейм) или суперкомпью-тер стоят дорого. Для достижения поставленных целей при проектировании

высокопроизводительных конструкций приходится иг-норировать стоимостные характеристики.

Суперкомпьютеры фирмы Cray Research и высокопроизводительные мейнфреймы компании IBM относятся именно к этой категории компьютеров. Другим крайним примером может служить низкостоимостная кон-струкция, где производительность принесена в жертву для дости-жения низкой стоимости. К этому направлению относятся персо-нальные компьютеры различных клонов IBM PC. Между этими двумя крайними направлениями находятся конструкции, основан-ные на отношении стоимость/ производительность , в которых раз-работчики находят баланс между стоимостными параметрами и производительностью. Типичными примерами такого рода ком-пьютеров являются миникомпьютеры и рабочие станции.

Для сравнения различных компьютеров между собой обычно используются стандартные методики измерения произво-дительности. Эти методики позволяют разработчикам и пользова-телям использовать полученные в результате испытаний количест-венные показатели для оценки тех или иных технических реше-

264

ний, и в конце концов именно производительность и стоимость дают пользователю рациональную основу для решения вопроса, какой компьютер выбрать.

**Надежность и отказоустойчивость.** Важнейшей ха-

рактеристикой вычислительных систем является надежность. По-вышение надежности основано на принципе предотвращения не-исправностей путем снижения интенсивности отказов и сбоев за счет применения электронных схем и компонентов с высокой и сверхвысокой степенью интеграции, снижения уровня помех, об-легченных 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, воз-можность 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. Кон-цепции параллельности и отказоустойчивости вычислительных систем естественным образом связаны между собой, поскольку в обоих случаях требуются дополнительные функциональные ком-поненты. 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 ис-пользоваться как для повышения производительности, так и для повышения надежности. Структура многопроцессорных и много-машинных систем приспособлена к автоматической реконфигура-ции 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 целост-ность хранимых в них данных.

**Масштабируемость**.Масштабируемость представляетсобой возможность наращивания числа и мощности процессоров,

265

объемов оперативной и внешней памяти и других ресурсов вычислительной системы. Масштабируемость должна обеспечи-ваться архитектурой и конструкцией компьютера, а также соот-ветствующими средствами программного обеспечения.

Добавление каждого нового процессора в действительно масштабируемой системе должно давать прогнозируемое увеличе-ние производительности и пропускной способности при приемле-мых затратах. Одной из основных задач при построении масшта-бируемых 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 производи-тельности могут возникать, например, при недостаточной пропу-скной способности шин из-за возрастания трафика между процес-сорами и основной памятью, а также между памятью и устройст-вами ввода/вывода. В действительности реальное увеличение про-изводительности трудно оценить заранее, поскольку оно в значи-тельной степени зависит от динамики поведения прикладных за-дач.

Возможность масштабирования системы определяется не только архитектурой аппаратных средств, но зависит от заложен-ных свойств программного обеспечения. Масштабируемость про-граммного обеспечения затрагивает все его уровни от простых ме-ханизмов передачи сообщений до работы с такими сложными объ-ектами 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 сред-ства (процессоры, шины и устройства ввода/ вывода) являются только частью масштабируемой архитектуры, на которой про-граммное обеспечение может обеспечить предсказуемый рост производительности. Важно понимать, что простой переход, на-пример , на более мощный процессор может привести к перегрузке других компонентов системы. Это означает, что действительно масштабируемая система должна быть сбалансирована по всем па-раметрам.

266



**Совместимость и мобильность программного обеспе-**

**чения .** Концепция программной совместимости впервые в ши-роких масштабах была применена разработчиками системы IBM/360. Основная задача при проектировании всего ряда моде-лей этой системы заключалась в создании такой архитектуры, ко-торая была бы одинаковой с точки зрения пользователя для всех моделей системы независимо от цены и производительности ка-ждой из них. Огромные преимущества такого подхода , позво-ляющего сохранять существующий задел программного обеспе-чения при переходе на новые (как правило, более производитель-ные) модели были быстро оценены как производителями компь-ютеров, так и пользователями и начиная с этого времени практи-чески все фирмы-поставщики компьютерного оборудования взя-ли 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 информацион-ных 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, по-зволяющее решить его задачи, а не выбор той или иной аппарат-

267

ной платформы. Переход от однородных сетей программно со-вместимых 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 инте-грации отдельных ресурсов - мощную распределенную вычисли-тельную систему, каждый элемент которой (сервер или рабочая станция) лучше всего соответствует требованиям конкретной при-кладной задачи.

Этот переход выдвинул ряд новых требований. Прежде всего такая вычислительная среда должна позволять гибко менять количество и состав аппаратных средств и программного обеспе-чения в соответствии с меняющимися требованиями решаемых за-дач. Во- вторых, она должна обеспечивать возможность запуска одних и тех же программных систем на различных аппаратных платформах, т.е. обеспечивать мобильность программного обеспе-чения. 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 раз-личные компоненты вычислительной среды, предназначенных для обеспечения мобильности программных средств в рамках неодно-родной, распределенной вычислительной системы.

Одним из вариантов моделей открытой среды является мо-

дель OSE (Open System Environment), предложенная комитетом

IEEE POSIX. На основе этой модели национальный институт стан-дартов 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 реко-

мендуемые для федеральных учреждений США спецификации в области информационных технологий, обеспечивающие мобиль-ность системного и прикладного программного обеспечения. Все ведущие производители компьютеров и программного обеспече-ния в США в настоящее время придерживаются требований этого документа.

268

**9.2. Классификация систем параллельной обработки данных**

На протяжении всей истории развития вычислительной техники делались попытки найти какую-то общую классифика-цию, под которую подпадали бы все возможные направления раз-вития компьютерных архитектур. Ни одна из таких классификаций не могла охватить все разнообразие разрабатываемых архитектур-ных решений и не выдерживала испытания временем. Тем не ме-нее в научный оборот попали и широко используются ряд терми-нов, которые полезно знать не только разработчикам, но и пользо-вателям компьютеров.

Любая вычислительная система (будь то супер-ЭВМ или персональный компьютер) достигает своей наивысшей производи-тельности 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 классифика-ции 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 при-знаки, которые в этой классификации не учтены. В частности , к машинам типа SIMD часто относят векторные процессоры, хотя их высокая производительность зависит от другой формы паралле-лизма - конвейерной организации машины. Многопроцессорные векторные системы, типа Cray Y- MP, состоят из нескольких век-торных процессоров и поэтому могут быть названы MSIMD (Multiple SIMD).

Классификация Флинна не делает различия по другим важ-ным для вычислительных моделей характеристикам, например, по уровню "зернистости" параллельных вычислений и методам син-хронизации.

Можно выделить четыре основных типа архитектуры сис-тем параллельной обработки:

269

* **конвейерная и векторная обработка.** Основу конвей-

ерной 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 сту-пенях 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 соот-ветствующим элементам пары векторов). Для настройки конвейера на выполнение конкретной операции может потребоваться неко-торое установочное время, однако затем операнды могут посту-пать в конвейер с максимальной скоростью, допускаемой возмож-ностями памяти. При этом не возникает пауз ни в связи с выбор-кой новой команды, ни в связи с определением ветви вычислений при условном переходе. Таким образом, главный принцип вычис-лений на векторной машине состоит в выполнении некоторой эле-ментарной операции или комбинации из нескольких элементарных операций, которые должны повторно применяться к некоторому блоку данных. Таким операциям в исходной программе соответст-вуют небольшие компактные циклы.

* **системы типа SIMD**.Машины типаSIMDсостоят избольшого 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 рабо-ту. Модель вычислений для машины SIMD очень похожа на мо-дель вычислений для векторного процессора: одиночная операция выполняется над большим блоком данных.

270

В отличие от ограниченного конвейерного функциони-рования векторного процессора, матричный процессор (синоним для большинства SIMD-машин) может быть значительно более гибким. Обрабатывающие элементы таких процессоров - это уни-версальные программируемые ЭВМ , так что задача, решаемая па-раллельно, может быть достаточно сложной и содержать ветвле-ния. Обычное проявление этой вычислительной модели в исход-ной программе примерно такое же, как и в случае векторных опе-раций: циклы на элементах массива, в которых значения, выраба-тываемые на одной итерации цикла, не используются на другой итерации цикла.

Модели вычислений на векторных и матричных ЭВМ на-столько схожи, что эти ЭВМ часто обсуждаются как эквивалент-ные.

- **системы типа** **MIMD**. Термин "мультипроцессор" покры-вает 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 муль-типроцессорной системе каждый процессорный элемент (ПЭ) вы-полняет свою программу достаточно независимо от других про-цессорных элементов.

Процессорные элементы, конечно, должны как-то связы-ваться 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 свя-зываются с помощью общей шины или сети обмена. В противопо-ложность этому варианту в слабосвязанных многопроцессорных системах (машинах с локальной памятью) вся память делится ме-жду процессорными элементами и каждый блок памяти доступен только связанному с ним процессору. Сеть обмена связывает про-цессорные элементы друг с другом.

Базовой моделью вычислений на MIMD- мультипроцессоре является совокупность независимых процессов, эпизодически об-ращающихся к разделяемым данным. Существует большое коли-чество вариантов этой модели. На одном конце спектра - модель распределенных вычислений, в которой программа делится на до-

271

вольно большое число параллельных задач, состоящих из мно-жества подпрограмм. На другом конце спектра - модель потоко-вых вычислений , в которых каждая операция в программе может рассматриваться как отдельный процесс. Такая операция ждет своих входных данных (операндов), которые должны быть пере-даны ей другими процессами. По их получении операция выпол-няется, 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 многопро-цессорные системы, в которых в качестве процессоров использу-ются 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 конст-рукции, которые позволяют программисту описывать "крупнозер-нистый" параллелизм. В пределах каждой задачи компилятор ав-томатически векторизует подходящие циклы. Машины типа MSIMD, как можно себе представить, дают возможность исполь-зовать лучший из этих двух принципов декомпозиции: векторные операции ("мелкозернистый" параллелизм) для тех частей про-граммы, которые подходят для этого, и гибкие возможности MIMD-архитектуры для других частей программы.

Многопроцессорные системы за годы развития вычисли-тельной техники претерпели ряд этапов своего развития. Истори-чески первой стала осваиваться технология SIMD. Однако в на-стоящее время наметился устойчивый интерес к архитектурам MIMD. Этот интерес главным образом определяется двумя факто-рами:

* архитектура MIMD дает большую гибкость: при наличии адекватной поддержки со стороны аппаратных средств и про-граммного обеспечения MIMD может работать как однопользова-тельская система, обеспечивая высокопроизводительную обработ-ку данных для одной прикладной задачи, как многопрограммная

272

машина , выполняющая множество задач параллельно, и как не-которая комбинация этих возможностей.

- архитектура MIMD может использовать все преимущества современной микропроцессорной технологии на основе строгого учета соотношения стоимость/производительность. В действи-тельности практически все современные многопроцессорные сис-темы строятся на тех же микропроцессорах, которые можно найти в персональных компьютерах, рабочих станциях и небольших од-нопроцессорных серверах.

Одной из отличительных особенностей многопроцессорной вычислительной системы является сеть обмена, с помощью кото-рой процессоры соединяются друг с другом или с памятью. Мо-дель 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 со-общений, другая - на использовании общей памяти.

* + многопроцессорной системе с общей памятью один про-цессор осуществляет запись в конкретную ячейку, а другой про-цессор 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".
  + архитектурах с локальной памятью непосредственное разделение памяти невозможно. Вместо этого процессоры полу-чают доступ к совместно используемым данным посредством пе-редачи сообщений по сети обмена. Эффективность схемы комму-никаций зависит от протоколов обмена, основных сетей обмена и пропускной способности памяти и каналов обмена.

Часто, и притом необосновано, в машинах с общей памятью

* векторных машинах затраты на обмен не учитываются, так как проблемы обмена в значительной степени скрыты от программи-ста. Однако накладные расходы на обмен в этих машинах имеются
* 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,

273

что приводит к состоянию насыщения. Модель системы с об-щей памятью очень удобна для программирования и иногда рас-сматривается как высокоуровневое средство оценки влияния об-мена на работу системы , даже если основная система в действи-тельности реализована с применением локальной памяти и прин-ципа передачи сообщений.

В сетях с коммутацией каналов и в сетях с коммутацией па-кетов по мере возрастания требований к обмену следует учитывать возможность перегрузки сети. Здесь межпроцессорный обмен свя-зывает 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 распадают-ся на два основных класса в зависимости от количества объеди-няемых 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 (обыч-но менее 32) процессоров. Сравнительно небольшое количество процессоров в таких машинах позволяет иметь одну централизо-ванную общую память и объединить процессоры и память с по-мощью одной шины. При наличии у процессоров кэш-памяти дос-таточного объема высокопроизводительная шина и общая память могут удовлетворить обращения к памяти, поступающие от не-скольких 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 сис-темы с распределенной памятью . Для того чтобы поддерживать большое количество процессоров приходится распределять основ-ную память между ними , в противном случае полосы пропускания памяти просто может не хватить для удовлетворения запросов, по-ступающих от очень большого числа процессоров. Естественно

274

при таком подходе также требуется реализовать связь процес-соров между собой. На рис. 67 показана структура такой системы.

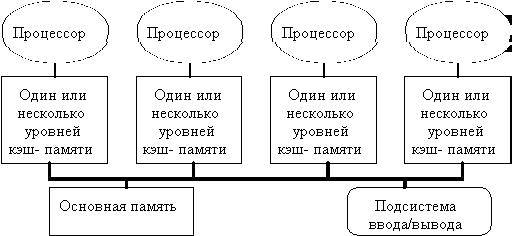


Рис. 66

С ростом числа процессоров просто невозможно обойти не-обходимость реализации модели распределенной памяти с высо-коскоростной сетью для связи процессоров. С быстрым ростом производительности процессоров и связанным с этим ужесточени-ем 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 па-мяти. Эти два преимущества еще больше сокращают количество процессоров, для которых архитектура с распределенной памятью имеет смысл.

275

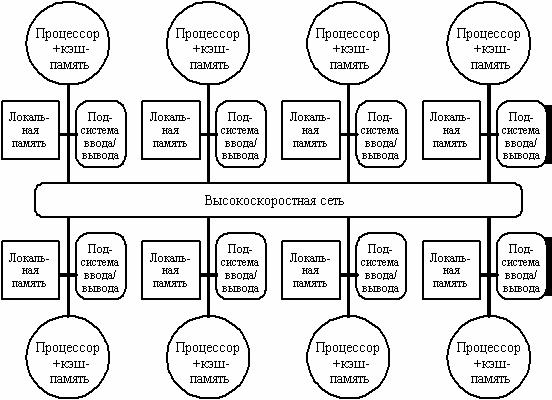


Рис. 67

Обычно устройства ввода/вывода, также как и память, рас-пределяются по узлам и в действительности узлы могут состоять из небольшого числа (2-8) процессоров, соединенных между собой другим способом. Хотя такая кластеризация нескольких процессо-ров с памятью и сетевой интерфейс могут быть достаточно полез-ными 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 про-цессором на узел. Основная разница в архитектуре, которую сле-дует выделить в машинах с распределенной памятью заключается в том, как осуществляется связь и какова логическая модель памя-ти.

276

**9.3. Модели связи и архитектуры памяти**

Как уже было отмечено, любая крупномасштабная много-процессорная система должна использовать множество устройств памяти, которые физически распределяются вместе с процессора-ми. Имеется две альтернативных организации адресации этих уст-ройств памяти и связанных с этим два альтернативных метода для передачи данных между процессорами. Физически отдельные уст-ройства памяти могут адресоваться как логически единое адресное пространство, что означает, что любой процессор может выпол-нять 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 раз-

деляемой памятью, а иногда NUMA's - Non-Uniform Memory Access, поскольку время доступа зависит от расположения ячейки

* памяти.
  + альтернативном случае, адресное пространство состоит из отдельных адресных пространств, которые логически не связа-ны 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 свя-зан свой механизм обмена. Для машины с единым адресным про-странством это адресное пространство может быть использовано для обмена данными посредством операций загрузки и записи. По-этому 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 на-зывают машинами с передачей сообщений.

Каждый из этих механизмов обмена имеет свои преимуще-ства. Для обмена в общей памяти это включает:

* + - совместимость с хорошо понятными используемыми как
* однопроцессорных, так и маломасштабных многопроцессорных

277

системах, механизмами, которые используют для обмена об-щую память.

* простота программирования, когда модели обмена между процессорами сложные или динамически меняются во время вы-полнения. 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, допус-кающая кэширование всех данных как разделяемых, так и нераз-деляемых.

Основные преимущества обмена с помощью передачи со-общений являются:

* аппаратура может быть более простой, особенно по срав-нению 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 надстроена над аппаратной моделью, которая использует любой из этих меха-низмов. Поддержка передачи сообщений над разделяемой памя-тью, естественно, намного проще, если предположить, что маши-ны имеют адекватные полосы пропускания. Основные трудности возникают при работе с сообщениями, которые могут быть непра-вильно выровнены и сообщениями произвольной длины в системе памяти, которая обычно ориентирована на передачу выровненных блоков данных, организованных как блоки кэш-памяти. Эти труд-ности можно преодолеть либо с небольшими потерями производи-тельности программным способом, либо существенно без потерь при использовании небольшой аппаратной поддержки.

Построение механизмов реализации разделяемой памяти над механизмом передачи сообщений намного сложнее. Без пред-полагаемой поддержки со стороны аппаратуры все обращения к разделяемой памяти потребуют привлечения операционной систе-мы как для обеспечения преобразования адресов и защиты памяти,

278

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 являют-ся три характеристики производительности:

* 1. *Полоса пропускания:* в идеале полоса пропускания меха-низма обмена будет ограничена полосами пропускания процессо-ра, памяти и системы межсоединений, а не какими-либо аспектами механизма обмена. Связанные с механизмом обмена накладные расходы (например, длина межпроцессорной связи) прямо воздей-ствуют на полосу пропускания.
  2. *Задержка:* в идеале задержка должна быть настолькомала, насколько это возможно. Для ее определения критичны на-кладные расходы аппаратуры и программного обеспечения, свя-занные с инициированием и завершением обмена.
  3. *Упрятывание задержки:* насколько хорошо механизмскрывает задержку путем перекрытия обмена с вычислениями или
* другими обменами.

Каждый из этих параметров производительности воздейст-вует 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 сообще-ний, поскольку это было очевидно проще и многие разработчики и исследователи не верили, что единое адресное пространство мож-но построить и в машинах с распределенной памятью. С недавнего времени модели обмена с общей памятью действительно начали поддерживаться практически в каждой разработанной машине (ха-рактерным примером могут служить системы с симметричной мультипроцессорной обработкой).

279

Хотя машины с централизованной общей памятью, по-строенные на базе общей шины все еще доминируют в терминах размера компьютерного рынка, долговременные технические тен-денции направлены на использование преимуществ распределен-ной памяти даже в машинах умеренного размера. Как мы увидим, возможно наиболее важным вопросом, который встает при созда-нии машин с распределенной памятью, является вопрос о кэширо-вании 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 тре-бования 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 разде-ляемой шины.

Из-за малого размера процессоров и заметного сокращения требуемой полосы пропускания шины, достигнутого за счет воз-можности реализации достаточно большой кэш-памяти, такие ма-шины стали исключительно эффективными по стоимости. В пер-вых разработках подобного рода машин удавалось разместить весь процессор и кэш на одной плате, которая затем вставлялась в зад-нюю панель, с помощью которой реализовывалась шинная архи-тектура. Современные конструкции позволяют разместить до че-тырех процессоров на одной плате. В такой машине кэши могут содержать как разделяемые, так и частные данные. Частные дан-ные - это данные, которые используются одним процессором, в то время как разделяемые данные используются многими процессо-рами, по существу обеспечивая обмен между ними. Когда кэширу-ется элемент частных данных, их значение переносится в кэш для сокращения среднего времени доступа, а также требуемой полосы пропускания.

280

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 требуе-мой полосы пропускания такая репликация данных способствует также общему сокращению количества обменов. Однако кэширо-вание разделяемых данных вызывает новую проблему: когерент-ность кэш-памяти.

**Мультипроцессорная когерентность кэш-памяти.** Про-

блема, о которой идет речь, возникает из-за того, что значение элемента данных в памяти, хранящееся в двух разных процессо-рах, 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, называе-мый протоколом, позволяющий решить эту проблему. Такие про-токолы называются протоколами когерентности кэш-памяти.

Существуют два класса таких протоколов:

* 1. Протоколы на основе справочника (directory based). Ин-формация о состоянии блока физической памяти содержится толь-ко в одном месте, называемом справочником (физически справоч-ник может быть распределен по узлам системы). Этот подход бу-дет рассмотрен в разд. 10.3.
  2. Протоколы наблюдения (snooping). Каждый кэш, кото-рый содержит копию данных некоторого блока физической памя-ти, имеет также соответствующую копию служебной информации
* 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 соответствующе-го блока.

281

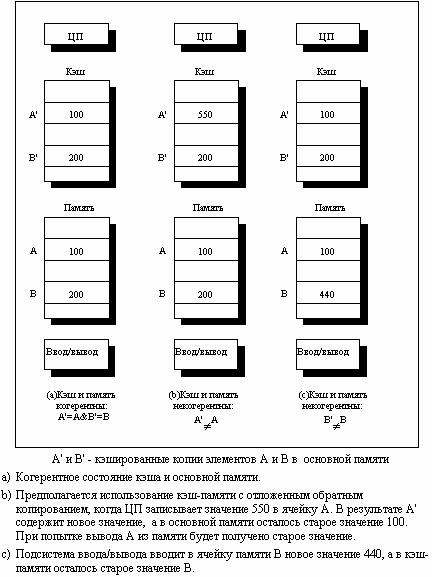


Рис. 68

282

В мультипроцессорных системах, использующих мик-ропроцессоры с кэш-памятью, подсоединенные к централизован-ной общей памяти, протоколы наблюдения приобрели популяр-ность, поскольку для опроса состояния кэшей они могут использо-вать заранее существующее физическое соединение - шину памя-ти.

Неформально, проблема когерентности памяти состоит в необходимости гарантировать , что любое считывание элемента данных возвращает последнее по времени записанное в него зна-чение. Это определение не совсем корректно, поскольку невоз-можно требовать, чтобы операция считывания мгновенно видела значение, записанное в этот элемент данных некоторым другим процессором. Если , например, операция записи на одном процес-соре предшествует операции чтения той же ячейки на другом про-цессоре в пределах очень короткого интервала времени, то невоз-можно гарантировать, что чтение вернет записанное значение дан-ных, поскольку в этот момент времени записываемые данные мо-гут даже не покинуть процессор.

Вопрос о том, когда точно записываемое значение должно быть доступно процессору, выполняющему чтение, определяется выбранной моделью согласованного (непротиворечивого) состоя-ния памяти и связан с реализацией синхронизации параллельных вычислений. Поэтому с целью упрощения предположим, что мы требуем только, чтобы записанное операцией записи значение бы-ло 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 обеспече-ния двух свойств:

* 1. Операция чтения ячейки памяти одним процессором, ко-торая следует за операцией записи в ту же ячейку памяти другим процессором, получит записанное значение, если операции чтения
* записи достаточно отделены друг от друга по времени.
  1. Операции записи в одну и ту же ячейку памяти выполня-ются строго последовательно (иногда говорят, что они сериализо-ваны): это означает, что две подряд идущие операции записи в од-

283

ну и ту же ячейку памяти будут наблюдаться другими процес-сорами именно в том порядке, в котором они появляются в про-грамме процессора, выполняющего эти операции записи.

Первое свойство очевидно связано с определением коге-рентного (согласованного) состояния памяти : если бы процессор всегда бы считывал только старое значение данных, мы сказали бы, что память некогерентна.

Необходимость строго последовательного выполнения опе-раций записи является более тонким, но также очень важным свойством. Представим себе, что строго последовательное выпол-нение 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 за-писи гарантирует два важных следствия для этой последователь-ности операций записи . Во-первых, оно гарантирует, что каждый процессор в машине в некоторый момент времени будет наблю-дать запись, выполняемую процессором P2. Если последователь-ность операций записи не соблюдается , то может возникнуть си-туация, когда какой-нибудь процессор будет наблюдать сначала операцию записи процессора P2, а затем операцию записи процес-сора P1, и будет хранить это записанное P1 значение неограничен-но 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 процес-соры 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 мо-жет наблюдать за значениями, записываемыми как P1, так и P2, должен наблюдать идентичное поведение. Простейший способ до-биться таких свойств заключается в строгом соблюдении порядка операций записи, чтобы все записи в одну и ту же ячейку могли

284

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 под-

держания описанной выше когерентности. Один из методов за-ключается в том, чтобы гарантировать, что процессор должен по-лучить исключительные права доступа к элементу данных перед выполнением записи в этот элемент данных. Этот тип протоколов называется протоколом записи с аннулированием (write ivalidate 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. Исключи-тельное право доступа гарантирует, что во время выполнения за-писи не существует никаких других копий элемента данных, в ко-торые можно писать или из которых можно читать: все другие кэ-шированные копии элемента данных аннулированы.

Чтобы увидеть, как такой протокол обеспечивает когерент-ность, 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 соответст-вии с названием протокола).

Таким образом, когда возникает операция чтения, произой-дет 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 выигра-ет 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 со-держать обновленное значение.

285

Альтернативой протоколу записи с аннулированием яв-ляется 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 запи-си с трансляцией (write broadcast protocol). Обычно в этом прото-коле для снижения требований к полосе пропускания полезно от-слеживать, является ли слово в кэш-памяти разделяемым объек-том, 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 характери-стиками:

1. Несколько последовательных операций записи в одно и то же слово, не перемежающихся операциями чтения, требуют не-скольких операций трансляции при использовании протокола за-писи с обновлением, но только одной начальной операции анну-лирования при использовании протокола записи с аннулировани-ем.
2. При наличии многословных блоков в кэш-памяти каждое слово, записываемое в блок кэша, требует трансляции при исполь-зовании протокола записи с обновлением, в то время как только первая запись в любое слово блока нуждается в генерации опера-ции аннулирования при использовании протокола записи с анну-лированием. Протокол записи с аннулированием работает на уровне блоков кэш-памяти, в то время как протокол записи с об-новлением должен работать на уровне отдельных слов (или бай-тов, если выполняется запись байта).
3. Задержка между записью слова в одном процессоре и чтением записанного значения другим процессором обычно мень-ше при использовании схемы записи с обновлением, поскольку за-писанные данные немедленно транслируются в процессор, выпол-няющий 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

286

чтение данных и его приостановка до тех пор, пока обновлен-ная 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 копировани-ем. Также как и схема задержанной записи с обратным копирова-нием требует меньшей полосы пропускания памяти, так как она использует преимущества операций над целым блоком, протокол записи с аннулированием обычно требует менее тяжелого трафи-ка, чем протокол записи с обновлением, поскольку несколько за-писей 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 бу-фере записи). Подобным образом при использовании протокола записи с обновлением другие копии обновляются так быстро, на-сколько это возможно. Наиболее важное отличие в производи-тельности протоколов записи с аннулированием и с обновлением связано с характеристиками прикладных программ и с выбором размера блока.

**Основы реализации.** Ключевым моментом реализации вмногопроцессорных системах с небольшим числом процессоров как схемы записи с аннулированием, так и схемы записи с обнов-лением данных, является использование для выполнения этих опе-раций механизма шины. Для выполнения операции обновления или аннулирования процессор просто захватывает шину и транс-лирует по ней адрес, по которому должно производиться обновле-ние или аннулирование данных.

Все процессоры непрерывно наблюдают за шиной, контро-лируя появляющиеся на ней адреса. Процессоры проверяют не на-ходится 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. После-довательный порядок обращений, присущий шине, обеспечивает также строго последовательное выполнение операций записи, по-скольку когда два процессора конкурируют за выполнение записи в одну и ту же ячейку, один из них должен получить доступ к ши-не раньше другого.

287

Один процессор , получив доступ к шине, вызовет необ-ходимость обновления или аннулирования копий в других процес-сорах. В любом случае, все записи будут выполняться строго по-следовательно. Один из выводов, который следует сделать из ана-лиза этой схемы, заключается в том, что запись в разделяемый элемент данных не может закончиться до тех пор, пока она не за-хватит доступ к шине.

В дополнение к аннулированию или обновлению соответст-вующих копий блока кэш-памяти, в который производилась за-пись, мы должны также разместить элемент данных, если при за-писи происходит промах кэш-памяти. В кэш-памяти со сквозной записью последнее значение элемента данных найти легко, по-скольку все записываемые данные всегда посылаются также и в память, из которой последнее записанное значение элемента дан-ных может быть выбрано (наличие буферов записи может привес-ти к некоторому усложнению).

Однако для кэш- памяти с обратным копированием задача нахождения последнего значения элемента данных сложнее, по-скольку 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 предпочти-тельнее в мультипроцессорах, несмотря на некоторое увеличение сложности. Поэтому далее мы рассмотрим вопросы реализации кэш-памяти с обратным копированием.

Для реализации процесса наблюдения могут быть исполь-зованы обычные теги кэш. Более того, упоминавшийся ранее бит достоверности (valid bit), позволяет легко реализовать аннулиро-вание. Промахи операций чтения, вызванные либо аннулировани-ем, либо каким-нибудь другим событием, также не сложны для понимания , поскольку они просто основаны на возможности на-блюдения. Для операций записи мы хотели бы также знать, име-

288

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 поло-су пропускания.

Чтобы отследить, является ли блок разделяемым, мы можем ввести дополнительный бит состояния (shared), связанный с каж-дым блоком , точно также как это делалось для битов достоверно-сти (valid) и модификации (modified или dirty) блока. Добавив бит состояния, определяющий является ли блок разделяемым, мы мо-жем решить вопрос о том, должна ли запись генерировать опера-цию аннулирования в протоколе с аннулированием, или операцию трансляции при использовании протокола с обновлением. Если происходит запись в блок, находящийся в состоянии "разделяе-мый" 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. Процес-сор с исключительной (exclusive) копией блока кэш-памяти обыч-но 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 аннули-рованием, когда посылается операция аннулирования, состояние блока меняется с " разделяемый" на " неразделяемый" (или "част-ный "). Позже, если другой процессор запросит этот блок, состоя-ние 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 адрес-ные теги кэша, потенциально это может приводить к конфликтам с обращениями к кэшу со стороны процессора. Число таких потен-циальных конфликтов можно снизить применением одного из двух методов: дублированием тегов, или использованием много-уровневых кэшей с "охватом " (inclusion), в которых уровни, нахо-дящиеся ближе к процессору являются поднабором уровней, нахо-дящихся дальше от него. Если теги дублируются, то обращения

289

процессора и наблюдение за шиной могут выполняться парал-лельно. Конечно, если при обращении процессора происходит промах, он должен будет выполнять арбитраж с механизмом на-блюдения для обновления обоих наборов тегов.

Точно также, если механизм наблюдения за шиной находит совпадающий тег, ему будет нужно проводить арбитраж и обра-щаться к обоим наборам тегов кэш (для выполнения аннулирова-ния или обновления бита "разделяемый"), возможно также и к массиву данных в кэше, для нахождения копии блока. Таким обра-зом, 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 ме-ханизма наблюдения задерживается только тогда, когда кэш име-ет дело с промахом.

Если процессор использует многоуровневый кэш со свойст-вами 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. Ес-ли 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 при-нято во многих современных системах, поскольку многоуровне-вый кэш позволяет существенно снизить требований к полосе пропускания. Иногда может быть даже полезно дублировать теги во вторичном кэш, чтобы еще больше сократить количество кон-фликтов между активностями процессора и механизма наблюде-ния.

В реальных системах существует много вариаций схем ко-герентности 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

290

с наблюдением и некоторые машины, которые используют эти протоколы.

Таблица 8

|  |  |  |  |
| --- | --- | --- | --- |
| Наимено- |  | Тип |  |
| вание |  | протокола |  |
|  |  |  |  |
| Одиночная |  | Запись с ан- |  |
| запись |  | нулировани- |  |
|  | ем |  |
|  |  |  |
|  |  |  |  |
| Synapse |  | Запись с ан- |  |
|  | нулировани- |  |
| N+1 |  |  |
|  | ем |  |
|  |  |  |
|  |  |  |  |

Стратегия записи в память

Обратное копирование при пер-вой записи

Обратное

копирование

Уникальные свойства При-менение

Первый описанный в лите-ратуре протокол наблюде-ния -

Точное состояние , где "вла-дельцем является память" Машины Synapse. Первые машины с когерентной кэш-памятью

Berkely

Illinois

"Firefly"

Запись с ан-нулировани-ем

Запись с ан-нулировани-ем

Запись с транс-ляцией

Обратное

копирование

Обратное

копирование

Обратное копирова-ние для "приватных" блоков и сквозная за-пись для "разделяе-мых"

Состояние "разделяемый" Машина SPUR университета Berkely

Состояние "приватный"; может передавать данные из любого кэша Серии Power и Challenge компании Silicon Graphics

Обновление памяти во вре-мя трансляции SPARCcenter 2000

**9.5. Многопроцессорные системы с локальной памятью**

Существуют два различных способа построения крупно-масштабных систем с распределенной (локальной) памятью. Про-стейший способ заключается в том, чтобы исключить аппаратные механизмы, обеспечивающие когерентность кэш -памяти, и сосре-доточить 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

291

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, нахо-дятся ли требуемые данные в локальной памяти данного узла , или размещаются в памяти удаленного узла . В последнем случае кон-троллеру удаленной памяти посылается сообщение для обращения к требуемым данным.

Чтобы обойти проблемы когерентности, разделяемые (об-щие) 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 обес-печение. Преимуществом такого подхода является практически минимальная необходимая поддержка со стороны аппаратуры, хо-тя наличие, например, таких возможностей как блочное (группо-вое) копирование данных было бы весьма полезным. Недостатком такой организации является то, что механизмы программной под-держки 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 параллелиз-

мом 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. Происшедшее воз-растание нормы массивности для больших машин до 512 и даже 64К процессоров обусловлено не ростом размеров машины, а по-вышением степени интеграции схем, позволившей за последние годы резко повысить плотность размещения элементов в устройст-вах. Топология сети обмена между процессорами в такого рода системах может быть различной. В таблице 9 приведены характе-ристики сети обмена для некоторых коммерческих MPP.

Таблица 9

292

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Фирма |  | Название | Количество |  | Базовая |  |
|  |  |  | узлов |  | 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 |  | T3D | 16-1024 |  | 3-мерный тор |  |
| Research |  |  |  |
|  |  |  |  |  |  |

Для построения крупномасштабных систем альтернативой рассмотренному в предыдущем разделе протоколу наблюдения может служить протокол на основе справочника, который отсле-живает состояние кэшей . Такой подход предполагает, что логиче-ски единый справочник хранит состояние каждого блока памяти, который может кэшироваться. В справочнике обычно содержится информация о том, в каких кэш имеются копии данного блока, мо-дифицировался ли данный блок и т.д. В существующих реализа-циях 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 прото-кола, связанная с тем, что вся необходимая информация сосредо-точена в одном месте. Недостатком такого рода справочников яв-ляется его размер, который пропорционален общему объему памя-ти, а не размеру кэш-памяти. Это не составляет проблемы для ма-шин, состоящих, например, из нескольких сотен процессоров, по-скольку связанные с реализацией такого справочника накладные расходы можно преодолеть. Но для машин большего размера не-обходима методика, позволяющая эффективно масштабировать структуру справочника.

293

В частности, чтобы предотвратить появление узкого гор-ла в системе, связанного с единым справочником, можно распре-делить части этого справочника вместе с устройствами распреде-ленной локальной памяти. Таким образом можно добиться того, что обращения к разным справочникам (частям единого справоч-ника) 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 свой-ство подобных схем , заключающееся в том, что состояние любого разделяемого блока данных всегда находится во вполне опреде-ленном известном месте.

На рис. 69 показан общий вид подобного рода машины с распределенной памятью. Вопросы детальной реализации прото-колов когерентности памяти для таких машин выходят за рамки настоящего обзора.

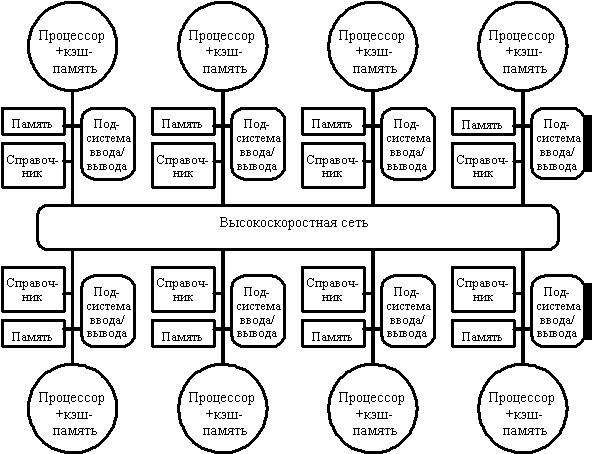


Рис. 69

294

**10. Режимы обмена в МПС**

Организация обмена в МПС - важная функция микропроцес-сора. В обмене принимают участие активное и пассивные устрой-ства. Активным устройством в большинстве случаев является МП, пассивным - основная память или ВУ.

Основными режимами обмена являются программно - управ-ляемый обмен, обмен в режиме прерывания и обмен в режиме пря-мого доступа к памяти.

**Программно -управляемый обмен.** Обмен осуществляетсяпо 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 ини-циативе обрабатываемой команды и включает чтение информа-ции 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 за-писывается 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 со-единен с требуемым ВУ и через который будет происходить ввод (чтение) информации в МП. Такой канал называется портом. Че-рез некоторое время на линии ШУ МП формирует управляющий

295

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.

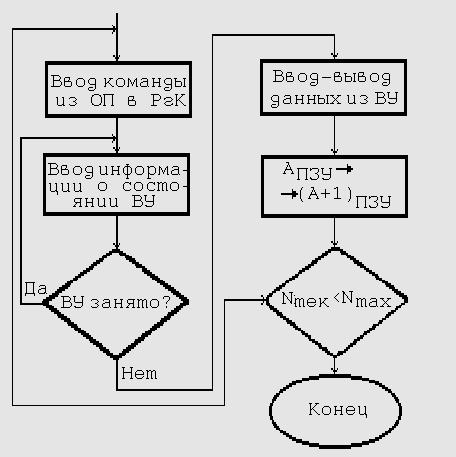


Рис. 70.

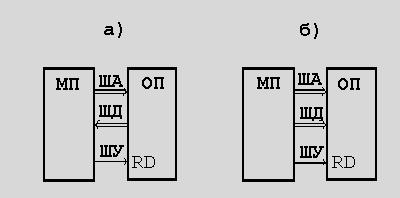


Рис. 71.

296

Вывод информации из МП во внешнее устройство осу-ществляется путем формирования МП на линиях ША адреса ка-нала (порта) КВВ, который соединен с требуемым ВУ. Через не-который промежуток времени МП формирует на линии ШУ сиг-нал WR «запись » (или «вывод») и выдает на ШД информацию, которая должна быть записана (выведена) в требуемое ВУ. За-прошенный 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 аккуму-лятором : 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 уча-стием МП не ставился вопрос о готовности пассивных ВУ к про-ведению обмена. Между тем их быстродействие существенно раз-личается 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 об-мену. Обмен в этом случае носит название синхронным обменом. В тех случаях, когда быстродействие ВУ ниже быстродействия МП (например, клавиатура пульта оператора), то синхронный спо-соб обмена неприменим и используется так называемый асин-хронный обмен.

Асинхронный обмен происходит также под управлением программы, но лишь в том случае, когда ВУ подготовлено к обме-ну. Об этом сообщает сигнал готовности, формируемый ВУ через КВВ (или самим КВВ) на соответствующей линии ШУ. Процесс обмена, инициированный программой, начинается с анализа про-цессором готовности ВУ к обмену. При отсутствии сигнала готов-ности МП переходит в состояние ожидания, о чем извещает ос-тальные функциональные модули специальным сигналом (WA*I*T) на одной из линий ШУ. После прихода сигнала готовности проис-ходит непосредственная процедура обмена.

Основным недостатком асинхронного обмена являются по-тери времени процессора на ожидание того момента, когда уст-ройство будет готово к обмену, и такие потери для некоторых уст-ройств могут оказаться значительными. Так, например, при вводе

297

информации с пульта оператора среднее время между нажатия-ми 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 пре-вышает 0,01% общего времени обмена.

Из приведенного примера видно, что за время ожидания оче-редного сигнала готовности МП способен выполнить достаточно большое число операций в соответствии с командами программы, если бы сигнал готовности ВУ мог останавливать (прерывать) вы-полнение основной программы и переводить МП в режим выпол-нения процедуры обмена. Сигнал готовности такого вида называет-ся сигналом прерывания , а способ обмена с использованием сигна-лов прерывания получил название обмена в режиме прерывания.

**Обмен в режиме прерывания.** Обмен в режиме прерыва-

ния предназначен для обработки программ обслуживания запро-сов прерывания, сформированных ВУ в процессе накопления ими информации за время работы. Время формирования запросов пре-рывания ВУ - явление случайное и, в большинстве случаев, не мо-жет 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, вырабаты-вает сигнал запроса прерывания, который МП анализирует и, при необходимости, прерывает обрабатываемую программу и перехо-дит к операции обмена - вводу и обработке программы обслужи-вания прерывания.

Различают прерывания аппаратные, программные и специ-альные.

Аппаратные прерывания, на практике еще называемые внешними прерываниями, имеют место при воздействии сигналов, которые вырабатываются ВУ, требующими обслуживания. Аппа-ратные прерывания используются, как правило, для обслуживания ВУ по запросу этих устройств. Они могут быть немаскируемые и маскируемые.

298

Немаскируемые прерывания - прерывания, которые вы-зываются внешними, аппаратными средствами и не могут быть за-прещены выполняемой программой. Запросы на такие прерывания подаются на специальный вход микропроцессора - вход немаски-руемых прерываний. Они обслуживаются обязательно и немед-ленно вне зависимости от важности выполняемой в данный мо-мент 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 вхо-дов для запросов на обслуживание маскируемых прерываний.

Программные прерывания происходят под воздействием ко-манд прерывания, включенных в основную программу. Здесь ини-циатива программного прерывания исходит от самой программы. Программные прерывания используются:

для обслуживания устройств ввода - вывода по опросу, для вызова вспомогательных программ операционной сис-

темы - так называемых "утилит".

В последнем случае механизм прерываний оказывается более эффективным, чем механизм перехода на программу обслужива-ния прерывания, хотя в принципе они очень близки.

Специальные прерывания возникают в системе в ходе вы-полнения основной программы под воздействием сигналов, выра-батываемых 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 об-ратиться к запрещенной области памяти (при ошибке адресации),

* др.;

б) аппаратные сбои, являющиеся следствием критического изменения или внезапного отключения питания, неисправностей отдельных узлов аппаратуры, вызывающих неправильное функ-ционирование аппаратуры, и др.;

299

в) переполнение разрядной сетки, т. е. получение в ходе вычислений чисел, выходящих за пределы диапазона допустимых значений;

г) трассировка - выполнение программы в пошаговом режи-ме 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 отно-шению к приоритету системы и подключаются последовательно к линии, по которой передается сигнал разрешения прерывания (рис. 72). При этом ближайшему к МП контроллеру прерываний (КПр 1) присваивается наименьший адрес, а самому удаленному (КПрN) - наибольший.

Запросы от ВУ через соответствующие контроллеры преры-ваний КПр поступают на вход «1» запроса прерывания МП по единственной линии, к которой подключаются все КПр системы. При этом запросы на прерывание от одного или нескольких КПр могут маскироваться программно. Если на линии появляется за-прос от одного какого либо КПр, то МП воспринимает его, анали-зирует на предмет приоритетности ВУ, пославшего запрос, и, в случае более высокого приоритета устройства по сравнению с приоритетом системы, проводит дополнительные операции по со-хранению 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 КПр. Если приоритет МП, веду-щего обработку текущей программы, выше приоритета уст-ройств, то любой запрос прерывания им игнорируется. Если же приоритет МП ниже приоритета всех ВУ , то в ответ на любой за-прос прерывания МП подготавливается к переходу обработки про-граммы обслуживания прерываний и посылает сигнал на линию «2» разрешения прерывания. Этот сигнал последовательно прохо-дит через все контроллеры прерываний, начиная с самого ближне-го по линии связи к МП, и кончая тем, который послал сигнал прерывания. Этот КПр запрещает , или блокирует, дальнейшее его распространение и выдает на ШД начальный адрес программы об-

300

служивания прерывания, уникальный для данного ВУ. МП вос-воспринимает этот адрес, транслирует его передачу на ША, вводит первую команду программы обслуживания прерывания и начинает ее выполнение.

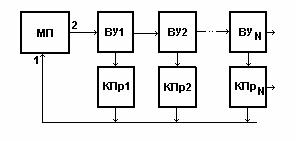


Рис .72.

Последовательность действий, выполняемая МП при обра-ботке прерывания от ВУ, которая справедлива также при обра-ботке 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 специаль-ной команде, с помощью указателя стека, МП запоминает в стеке сначала содержимое счетчика команд (т.е. адрес *i* - й команды об-рабатываемой программы), а затем текущее содержимое регистра состояния МП, которое считается «старым» словом состояния (ССП) МП, и, при необходимости, содержимое программно - дос-тупных регистров (например, аккумулятора, РОН), участвовавших в формировании результатов обработки прерванной программы, а также дополнительно проводит необходимые операции по обслу-живанию прерывания, зависящий от источника прерывания.

2. По адресу «вектора прерывания», сформированного кон-троллером прерываний, МП загружает из вектора прерывания в счетчик команд начальный адрес программы обслуживания пре-рывания, а в регистр состояний - «новое» слово состояния МП.

301

Вектор прерывания - это адрес первой команды програм-мы обслуживания определенного типа прерываний. В некоторых микропроцессорах, например, в МП Z80000 фирмы Zilog ( США), К1801ВМЗ, вектор включает в себя не только адрес первой коман-ды 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 обслу-живания 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 про-граммы обслуживания прерывания. Если есть необходимость в за-прете прерывания до полного завершения программы обслужива-ния прерывания, то в «новом» слове состояния МП на время об-работки программы обслуживания прерывания должен быть уста-новлен высший приоритет МП.

После своего завершения программа обслуживания прерыва-ния 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 про-граммно - 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 одно-временно послать запросы от ВУ на обслуживание прерывания. Очевидно , в МП должен быть предусмотрен механизм для приема в каждый момент времени единственного запроса для обслужива-ния прерывания. Этот механизм реализуется, как правило, аппа-ратными средствами и позволяет в первую очередь выбирать по-рядок обслуживания ВУ: либо первым удовлетворяется запрос прерывания от того контроллера, который включен ближе к МП в цепочке передачи сигналов «запрос прерывания - разрешение пре-

302

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, кото-рые могут прервать друг друга, характеризуют «уровень прерыва-ния». При этом порядок обработки программ при одновременном их запросе определяется уровнем приоритета обслуживаемых ВУ.

* многоуровневых системах прерывания задание уровня приоритета осуществляется тремя способами:

а) использованием в самом МП (или в контроллере прерыва-ний) нескольких входов запросов на прерывание, каждый из кото-рых имеет свой уровень приоритета;

б) с помощью 2 - 4-разрядной шины для подачи кода, несу-щего информацию об уровне приоритета того или иного КПр, за-просившего прерывание;

в) использованием внешних аппаратных средств.

* многоуровневых системах прерываний программа обслу-живания ВУ низкого уровня может быть прервана запросом на прерывание от Кпр, обслуживающего ВУ высокого уровня. Если же запрос на прерывание от КПр имеет тот ж уровень, что и об-служиваемое на момент запроса прерывание, или более низкий, то МП не реагирует на этот запрос до тех пор, пока не закончит об-служивание устройства Здесь условно показано, что после поступ-ления сигнала запроса прерывания управление запоминанием со-стояния и возвратом в прерываемую программу возложено на саму прерывающую программу. В этом случае прерывающая программа состоит из трех частей:

подготовительной, осуществляющей запоминание состоя-

ния прерванной программы, заключительной, обеспечивающей восстановление состоя-

ния прерванной программы,

303

прерывающей программы, выполняющей затребован-ный запросом режим. с более высоким уровнем приоритета.

Для оценки эффективности системы прерывания, реализую-щей обмен, используются следующие характеристики системы прерывания.

Время реакции - время между появлением запроса прерыва-ния и началом выполнения прерывающей программы.

На рис. 73 приведена упрощенная временная диаграмма про-цесса прерывания.

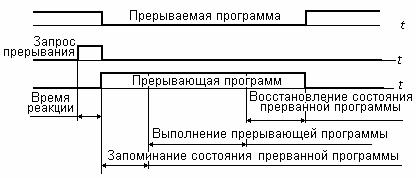


Рис. 73

Для одного и того же запроса прерывания задержки в испол-нении прерывающей программы зависят от того, сколько про-грамм с более высоким приоритетом ВУ ждут обслуживания . По-этому время реакции определяется для запроса с наивысшим при-оритетом ВУ.

Время реакции зависит от того, в какой момент допустимо прерывание. Большей частью прерывание допускается после окончания текущей команды. В этом случае время реакции опре-деляется 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 вы-полнения команды. Однако при этом возрастает количество ин-формации, подлежащей запоминанию и восстановлению при пере-

304

ключении программ, так как в этом случае необходимо сохра - нять также и состояния в момент прерывания счетчика тактов, ре-гистра кода операции и некоторых других. Поэтому такая органи-зация прерывания возможна в МПС с быстродействующей сверх-оперативной памятью.

Имеются ситуации, в которых желательно немедленное пре-рывание. Если аппаратура контроля обнаружила ошибку, то целе-сообразно сразу же прервать операцию, пока ошибка не оказала влияния на следующие такты работы системы.

Затраты времени на переключение программ равны суммар-ному расходу времени на запоминание и восстановление состоя-ния программы:

Глубина прерывания - максимальное число программ, кото-рые могут прерывать друг друга. Если после перехода к преры-вающей программе и вплоть до ее окончания прием других запро-сов запрещается, то система имеет глубину прерывания, равную «1». Если допускается последовательное прерывание до N про-грамм, то глубина прерывания будет равна N.

Глубина прерывания обычно совпадает с числом уровней приоритета в системе прерываний. На рис. 74 показан обмен в ре-жиме 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

305

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» обра-ботки предыдущего прерывания до ее завершения. Только после этого он приступает к обработке следующей программы обработ-ки прерывания.

Во втором случае обработка текущей программы прерывает-ся в случае , если на вход системы поступает сигнал запроса пре-рывания от устройства с более высоким приоритетом, чем приори-тет текущей программы. Так , например, сигнал запроса прерыва-ния «2» (рис . 74,б) ВУ более высокого приоритета, чем приоритет МПС, ведет к прерыванию текущей программы и началу обработ-ки программы обслуживания прерывания ВУ, выставившего за-прос «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 прерван-ную 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 не-обслуженным к моменту прихода нового запроса от того же ис-точника, то возникает так называемое насыщение системы преры-вания. В этом случае предыдущий запрос прерывания от данного источника будет системой утрачен, что является недопустимым. Для устранения этого явления быстродействие МПС, характери-стики системы прерывания, число источников прерывания и час-тота возникновения запросов должны быть согласованы таким об-разом, чтобы насыщение было невозможным.

Приоритетное обслуживание запросов прерывания. Вектор прерывания содержит всю необходимую информацию для пере-хода к прерывающей программе, в том числе ее начальный адрес. Каждому запросу (уровню) прерывания соответствует свой вектор прерывания, способный инициировать выполнение соответствую-щей прерывающей программы. Векторы прерывания обычно на-ходятся в специально выделенных фиксированных ячейках памя-ти. Главное место в процедуре перехода к прерывающей програм-ме занимают передача из соответствующего регистра (регистров)

306

процессора в память (в частности, в стек) на сохранение теку-щего вектора состояния прерываемой программы (чтобы можно было вернуться к ее исполнению) и загрузка в регистр (регистры) процессора вектора прерывания прерывающей программы, к ко-торой при этом переходит управление процессом.

Процедура организации перехода к прерывающей программе включает в себя выделение из выставленных запросов такого, ко-торый имеет наибольший приоритет.

Различают абсолютный и относительный приоритеты. За-прос, имеющий абсолютный приоритет, прерывает выполняемую программу и инициирует выполнение соответствующей преры-вающей программы. Запрос с относительным приоритетом явля-ется первым кандидатом на обслуживание после завершения вы-полнения текущей программы. Если наиболее приоритетный из выставленных запросов прерывания не превосходит по уровню приоритета выполняемую процессором программу, то запрос пре-рывания игнорируется или его обслуживание откладывается до за-вершения выполнения текущей программы.

Простейший способ установления приоритетных соотноше-ний между запросами (уровнями) прерывания состоит в том, что приоритет определяется порядком присоединения линии сигналов запросов ко входам системы прерывания. При появлении несколь-ких запросов прерывания первым воспринимается запрос, посту-пивший на вход с меньшим номером. В этом случае приоритет яв-ляется жестко фиксированным. Изменить приоритетные соотно-шения в этом случае можно лишь пересоединением линий сигна-лов 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 фор-мируется сигнал подтверждения прерывания, инициирующий вы-полняемую специальной программой или аппаратурой процедуру опроса регистра запросов прерывания для установления источни-ка, выставившего запрос прерывания наивысшего приоритета.

307

Более гибким и динамичным является векторное прерыва-ние, при котором исключается опрос источников прерывания.

Программно - управляемый приоритет прерывающих про-грамм. Относительная степень важности программ, их частота по-вторения, относительная степень срочности в ходе вычислитель-ного процесса могут меняться, требуя установления новых при-оритетных отношений. Поэтому во многих случаях приоритет ме-жду 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 прерываю-щих 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 следова-тельно, и обрабатываемой в данный момент процессором про-граммы) относительно приоритетов запросов источников преры-вания (в основном внешних устройств) . Порог прерывания задает-ся 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 пре-рывания, по которому начинается процедура прерывания.

Маска прерывания представляет собой двоичный код, разря-ды 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 те-кущей 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 пре-рывания. Каждая прерывающая программа может установить свою маску. При формировании маски сигналы логической «1» уста-

308

навливаются в разряды , соответствующие запросам (преры-вающим программами) с более высоким, чем у данной программы, приоритетом.

**Обмен в режиме прямого доступа к памяти.** В рассмот-

ренных ранее режимах обмен информацией осуществляется ме-жду МП и ОП или между МП и ВУ. Однако на практике часто возникает необходимость оперативного обмена информацией ме-жду 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 пе-редана из ВУ (или ОП) в аккумулятор МП, во втором цикле - ин-формация 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 считы-вание информации из памяти во внешнее устройство происходит непосредственно без участия МП. Такой вид обмена получил на-звание обмена в режиме прямого доступа к памяти (ПДП).

Для организации обмена применяется специальное управ-ляющее устройство - контроллер ПДП, который при обмене вы-полняет функции активного устройства, то есть устанавливает ад-рес 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 ре-жиме прерывания магистралью (шинами адреса, данных и управ-ления) 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 вы-полнения команды управляющее устройство МП переводит бу-ферные устройства его шин в режим с высоким выходным сопро-тивлением (высокоимпедансное состояние). МП при этом отклю-чается от магистралей и его управляющее устройство на специ-

309

альной линии управления формирует сигнал разрешения захва-та шин контроллеру ПДП и представляет магистраль в его распо-ряжение для передачи либо одного слова, либо целого массива информации.

При передаче одного слова контроллер ПДП занимает маги-страль 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 присут-ствует сигнал «запрос захвата шин». Это так называемый мульти-плексный обмен в режиме ПДП, при котором МП может продол-жать выполнение основной программы в режиме разделения вре-мени с процедурой обмена в режиме ПДП. При этом нет необхо-димости 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 ос-танавливает 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 предусмотре-но несколько регистров: регистры адреса, в которых находятся начальное и текущее значения адресов, регистр - счетчик переда-ваемых слов, регистр управления, регистр состояния и т. п. Обыч-но в составе контроллера ПДП предусматривается несколько ка-налов для подключения внешних устройств. В этом случае кон-троллер помимо прочего осуществляет арбитраж запросов на за-хват 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.

310

**11. Каналы передачи информации в МПС**

Работа МПС характеризуется интенсивным обменом инфор-мацией между их основными частями: МП, ОП, УВВ и мультиси-стемными средствами. Связь МП с ОП, УВВ требует нескольких каналов передачи информации - интерфейсов. В зависимости от функционального назначения интерфейсные схемы делятся на не-сколько уровней (рис. 75).

**Шинные интерфейсы 1 уровня** обеспечивают обмен ин-

формацией между всеми ( или основными) модулями микропро-цессорной системы. Выбор большинства шин этого уровня опре-деляется архитектурой микропроцессора, например интерфейс И-41(Multibus) используется для МПК серий К580, K1810, межмо-дульный параллельный интерфейс (МПИ) - для МПК серий К1801/1809, К1811, К581, K5S8 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 че-рез соответствующие контроллеры, с вычислительным ядром МПС. В связи с увеличением скоростей работы внешних уст-ройств, постоянным ростом их числа и необходимостью одновре-менного использования ВУ требования к пропускной способности системной магистрали постоянно растут. Поэтому их число велико и продолжает расти, что отражает, с одной стороны, постоянные 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 техниче-ские характеристики которых в сильной степени зависят от со-вместимости сопрягаемых компонентов.

311

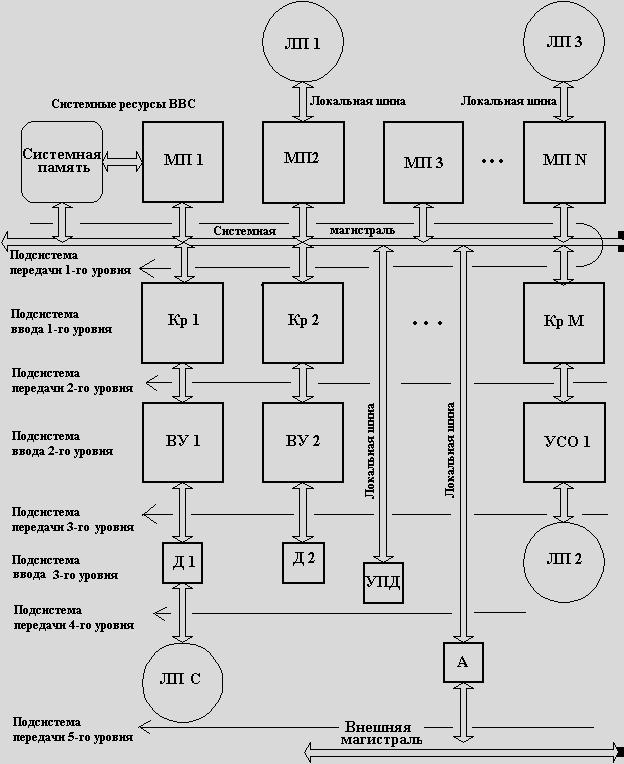


Рис. 75

Совместимость определяется следующими основными при-знаками: быстродействием, кодами, используемыми для обмена, архитектурой процессора, электрическими характеристиками. Ес-ли объединяемые компоненты не соответствуют друг другу по од-ному или нескольким признакам, то для взаимного подключения используют специальные электронные схемы, называемые интер-

312

фейсными модулями. Необходимость использования интер-фейсных модулей объясняется также и тем, что архитектура про-цессора с точки зрения набора и организации ШД, ША, набора управляющих сигналов определяет протокол или метод синхрони-зации МП и УВВ, и все передачи данных, кодов, признаков со-стояния, управляющих сигналов должны подчинятся этому прото-колу обмена. К тому же электрические характеристики МП долж-ны быть совместимы с характеристиками логических схем интер-фейса, которые в свою очередь согласуются с ВУ с помощью кон-троллеров.

Способы структурной и функциональной организации кон-троллеров 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 электриче-скую, информационную и конструктивную совместимость данного ВУ с системной магистралью , является сложной с технической и экономической точки зрения задачей. Поэтому наиболее рацио-нальным является стандартизация информационных и управляю-щих сигналов, которыми обмениваются МП с контроллером и ВУ.

С развитием микроэлектроники появилась возможность реа-лизовать стандартные интерфейсные функции в виде БИС. Для различных МПК БИС разработаны контроллеры, обеспечивающие связь ВУ по стандартному последовательному или по стандарт-ному параллельному каналу передачи данных.

Прежде, чем начать обзор шин, необходимо сказать несколь-ко 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 со-гласовать их работу. Основной обязанностью системной шины яв-ляется 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.

313

Таким образом, системную шину можно представить как совокупность сигнальных линий, объединенных по их назначе-нию:

* 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, раздельные 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 раз-дельные линии управления чтением и записью (фактически две та-кие линии используются для доступа к памяти, а две дополнитель-ных линии - для осуществления ввода- вывода). В этом случае центральный процессор посылает данные на периферийные уст-ройства, 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, присутст-вующие на реальной системной шине.

Существует множество системных шин, в том числе и ло-кальных, для PC и других типов компьютеров. Наиболее извест-

ными являются S-100, S-100 / IEEE696, ISA, EISA, Nubus, Mul-tibus-II, MCA, Sbus, Mbus, SCSI, VL-Bus, Futurebus+, VME, PCI.

**Шина S-100** была создана для8-разрядных микропроцес-соров и различных промышленных приложений. Типичные ее ха-рактеристики были такие:

* размеры: 134 мм x 254 мм, 100 выводов
* разъем: 50 выводов на каждой стороне платы
* нерегулируемое напряжение питания: +8В, +16В.

314

Шина S-100 нашла широкое применение в периферий-ных платах , она входила в состав плат памяти, устройств последо-вательного и параллельного интерфейсов , плат контроллеров гиб-ких магнитных дисков, видео- плат, плат музыкальных синтезато-ров 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 управ-ляющих 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 добав-ления еще двух сигналов sixteen request (SXTRO, 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 мо-дификации нашли применение при разработках небольших про-мышленных приложений. Основными достоинствами этой шины являются низкая цена и поддержка шины большим числом про-мышленных разработчиков.

У компьютеров IBM PC AT и IBM PC XT системная шина была предназначена для одновременной передачи только 8 разря-дов данных, так как используемый в компьютерах микропроцессор i8088 имел 8 линий данных. Кроме этого, системная шина включа-ла 20 адресных линий, которые ограничивали адресное простран-ство 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 (DМА - Direct Memory Access). Для подключения плат расширения

315

использовались специальные 62-контактные разъемы. Заметим , что системная шина и микропроцессор синхронизировались от од-ного тактового генератора с частотой 4,77 МГц. Таким образом, теоретически скорость передачи данных могла достигать более 4.5 Мбайт/с.

**Шина ISA.** В компьютерахPC AT,использующих микро-процессор i80286, впервые стала применяться новая системная шина ISA (Industry Standard Аrchitecture), полностью реализующая возможности упомянутого микропроцессора. Количество адрес-ных 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 приме-няться и в PC AT 286. Системные платы с шиной ISA позволили выполнять синхронизацию работы самой шины и микропроцессо-ра разными тактовыми частота ми, за счет чего устройства, выпол-ненные на платах расширения, могли работать медленнее, чем ба-зовый микропроцессор. Это стало особенно актуальным, когда тактовая частота процессоров превысила 10-12 МГц. Теперь сис-темная 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,

316

* 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 препятст-вий на пути повышения производительности компьютеров с этими микропроцессорами является системная шина типа ISA. Дело в том, что возможности этой шины для построения высокопроизво-дительных систем следующего поколения были практически ис-черпаны.

**Шина EISA** обеспечивает больший возможный объем адре-суемой памяти, 32-разрядную передачу данных, в том числе и в режиме DMA, улучшенную систему прерываний и арбитраж DMA, автоматическую конфигурацию системы и плат расшире-ния.

Шина EISA (Extended Industry Slandard Architecture) первона-

чально была ориентирована на вполне конкретную область приме-нения - на компьютеры, оснащенные высокоскоростными подсис-темами внешней памяти на жестких магнитных дисках с буферной кэш-памятью. Такие компьютеры до сих пор используются в ос-новном в качестве мощных файл-серверов или рабочих станций.

В EISA-разъем на системной плате компьютера, помимо, ра-зумеется, специальных EISA-плат, может вставляться либо 8-, ли-бо 16-разрядная плата расширения, предназначенная для обыкно-венной PC AT с шиной ISA. Это обеспечивается поистине гени-альным, 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". Бла-годаря этому сводится к минимуму вероятность генерации элек-тромагнитных помех, а также уменьшается восприимчивость к та-

317

ким помехам. Шина EISA позволяет адресовать 4-Гбайтное ад-ресное 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, орга-низующие режим DMA.

Стандарт EISA поддерживает многопроцессорную архитек-туру для "интеллектуальных" устройств (плат), оснащенных соб-ственными микропроцессорами. Поэтому данные, например, от контроллеров жестких дисков, графических контроллеров и кон-троллеров 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, из-вестных значений для ISA.

На шине EISA предусматривается метод централизованного управления, организованный через специальное устройство - сис-темный арбитр. Таким образом поддерживается использование ве-дущих устройств на шине, однако предусматривается также пре-доставление шины запрашивающим устройствам по циклическому принципу.

Как и для шины ISA, в системе EISA имеется 7 каналов DMA. Выполнение DMA- функций полностью совместимо с ана-логичными операциями на ISA- шине, хотя они могут происходить и несколько быстрее. Контроллеры DMA имеют возможность под-держивать 8 -, 16- и 32-разрядные режимы передачи данных. В об-щем случае возможно выполнение одного из четырех циклов об-мена между устройством DMA и памятью системы. Это - ISA - сoвмec тимые циклы, использующие для передачи данных 8 тактов шины; циклы типа A, исполняемые за 6 тактов шины; цикпы типа B, исполняемые за 4 такта шины, и циклы типа C (или burst), в ко-торых передача данных происходит за один такт шины. Типы цик-лов А, В и С поддерживаются 8-, 16- и 32-разрядными устройства-ми, причем возможно автоматическое изменение размера (шири-ны) данных при передаче в не соответствующую размеру память.

318

Большинство ISA-совместимых устройств, использующих DMA, могут работать почти в 2 раза быстрее, если они будут за-программированы на применение циклов А или В, а не стандарт-ных (и сравнительно медленных) ISA-циклов. Такая производи-тельность достигается только путем улучшения арбитража шины, а не в ущерб совместимости с ISA.

Приоритеты DMA в системе могут быть либо "вращающи-мися" (переменными), либо жестко установленными. Линии пре-рывания шины ISA, по которым запросы прерывания передаются в виде перепадов уровней напряжения (фронтов сигналов), сильно подвержены импульсным помехам. Поэтому в дополнение к при-вычным сигналам прерываний на шине ISA, активным только по своему фронту, в системе EISA предусмотрены также сигналы прерываний, активные по уровню. Причем для каждого прерыва-ния выбор той или иной схемы активности может быть запро-граммирован заранее. Собственно прерывания, активные по фрон-ту, 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 под-вержены шумам и помехам , нежели обычные. К тому же (теорети-чески) по одной и той же физической линии можно передавать бесконечно большое число уровней прерывания. Таким образом, одна линия прерывания может использоваться для нескольких за-просов.

Для компьютеров с шиной EISA предусмотрено автоматиче-ское конфигурирование системы. Каждый изготовитель плат рас-ширения для компьютеров с шиной EISA поставляет вместе с эти-ми 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 по-добрать правильное положение DIP-переключателей и перемычек, однако сервисная программа на EISA-компьютерах позволяет ото-бражать установленные положения соответствующих переключа-телей на экране монитора и дает некоторые рекомендации по пра-

319

вильной их установке. Помимо этого, в архитектуре EISA пре-дусматривается выделение определенных групп адресов ввода - вывода для конкретных слотов шины - каждому разъему расшире-ния отводится адресный диапазон 4Кбайта. Это также позволяет избежать конфликтов между отдельными платами EISA. Кроме то-го, шина по-прежнему тактируется частотой около 8 МГц, а ско-рость передачи увеличивается в основном благодаря увеличению разрядности шины данных. Отметим, что шина EISA имеет сле-дующие 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 в1985г.как развитиешироко применяемого в промышленной автоматике стандарта Multibus. Multibus-II является 32- разрядной шиной и может рабо-тать со скоростью управляющего процессора - вплоть до достиже-ния пропускной способности 80 Мбайт/с. В отличие от других рассматриваемых здесь шин, Multibus обладает возможностью вы-сокоскоростной передачи сообщений между различными управ-ляющими устройствами. При этом механизм передачи позволяет организовывать "интеллектуальное" взаимодействие между про-цессорами и контроллерами. Это особенно важно при создании многопроцессорных систем и построении сложных комплексов промышленной электроники.

Шина содержит пять магистралей (рис. 76), логическая орга-низация которых приспособлена для решения задач определенного класса:

i PSB - магистраль параллельная системная межмашин-

ная;

320

i LBX - магистраль параллельная локального расши-рения;

i SSB - магистраль последовательная системная;

i SBX - магистраль параллельная расширения ввода - вы-

вода;

MDMA - параллельная магистраль каналов прямого дос-тупа к памяти.

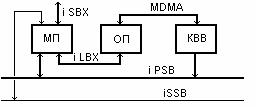


Рис. 76

Одновременное использование магистралей iPSB, iLBX, MDMA обеспечивает суммарную пропускную способность 96 Мбайт/с без учета скорости передачи по магистрали iSBX, пред-назначенной специально для организации ввода - вывода. Парал-лельные магистрали используют синхронное стробирование, по-вышающее помехоустойчивость. Магистрали iPSB и iLBX содер-жат контрольные разряды по четности, а магистраль iSSB исполь-зует 16 - разрядный код циклического избыточного кодирования.

Основная системная магистраль iPSB объединяет все модули системы и через нее осуществляется управление и контроль за ее работой. В ее состав входят 32 совмещенные линии адре-са/данных, пять линий параллельного арбитража, с помощью ко-торых осуществляется процедура приоритетов при прерываниях.

**Шина MC** (MicroChannel)появилась в1987г.в компьютерахPS/2. Достаточно быстрая (до 20 МГц, до 76 Мбайт/с) и широкая (32 бита), шина содержала рад удачных архитектурных решений и вполне могла бы бороться за лидерство среди системных шин. Она обладает следующими особенностями:

321

* + 8/16/32 - разрядные линии передачи данных,
  + прерывания по уровню сигнала (в отличие от ISA, где пре-рывания - по фронту синхросигнала),
  + 24 или 32 адресных линии (адресация до 4 Гбайт памяти),
  + автоматическая конфигурация плат (на основе информации
* ROM этих плат),
  + асинхронный протокол передачи данных.

**Шина Sbus** разработана в1989г.для работы с частотой до25 МГц. Она предназначена для передачи 32-разрядных данных. Ее особенностью являются возможность автоматически трансли-ровать виртуальные адреса в физические, распознавать ошибки при передаче данных и инициировать повторы.

**Шина Mbus** создана в1990г.и предназначена для передачи64-разрядных данных. Mbus допускает совместное использование с другими шинами, имеет портативные варианты исполнения и предусматривает возможности передачи сообщений.

**Шина SCSI** (Small Computer System Interface)регламентиро-

ван стандартом IEC 9316, который унифицирует основные уровни для базовых типов периферийных устройств, главным образом на-копителей магнитных дисков, АЦПУ, а также возможности рас-ширения функций посредством специальных кодов и полей. В ин-терфейсе используется логическая адресация всех блоков данных и возможность считывания с устройств прямого доступа информа-ции 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 приемопере-датчиков. Архитектура интерфейса предусматривает несколько видов организации взаимодействия эадатчиков (инициаторов) и исполнителей (приемников) с использованием необязательного распределенного арбитража. Время арбитража не превышает 10 мкс.

Дополнительные возможности такие: два варианта физиче-ской реализации, использование четности, синхронная передача данных и др. Команды разделены на обязательные, расширенные, необязательные и уникальные. Устройства выполняют все обяза-тельные команды для данного типа устройств команды, а также

322

ряд других команд. Кроме того. в стандарте определены расши - ренные команды дли устройств прямого доступа, постоянные ко-манды для всех типов устройств, уникальные команды для жест-ких дисков, ленточных накопителей, принтеров, оптических дис-ков, процессоров, байты состояния всех типов устройств.

Максимальное число подключенных устройств - 8. Каждое устройство идентифицируется соответствующим разрядом, раз-мещаемым 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, лен-точные накопители, сканеры, магнитооптические устрой-ства 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 ши-не. 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 переда-чи до 10 Мбайт/сек

Разработчики компьютеров , системные платы которых ос-новывались на микропроцессорах i80386/486, стали использовать раздельные шины для памяти и устройств ввода-вывода. Это по-зволило 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 производительно-сти, так как устройства, подключенные через разъемы расшире-

323

ния, не могут достичь скорости обмена, сравнимой с процессо-ром . В основном это касается работы с контроллерами накопите-лей и видеоадаптерами.

Для решения данной проблемы стали использовать так на-зываемые локальные (local или mezzanine) шины, которые непо-средственно 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, во-обще говоря, для одного и того же - для увеличения быстродейст-вия компьютера, позволяют таким периферийным устройствам, как видеоадаптеры и контроллеры накопителей, работать с такто-вой частотой 33 МГц и выше. Обе эти шины используют разъемы типа МСА.

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 пе-риферийного 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 мик-ропроцессора, может подключаться до трех периферийных уст-ройств. В качестве таких устройств в настоящее время выступают контроллеры накопителей, видеоадаптеры и сетевые платы.

Конструктивно VL-bus выглядит как короткий соединитель типа МСА (112 контактов), установленный, например, рядом с разъемами расширения ISA или EISA. При этом 32 линии исполь-зуются для передачи данных и 30 - для передачи адреса. Макси-мальная 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-

324

разрядный канал данных, буферизация сигналов для работы с быстродействующими системными платами и более высокая мак-симальная тактовая частота - 50 МГц. Количество разъемов рас-ширения увеличится до трех разъемов на 40 МГц и до двух на 50 МГц. Ожидаемая скорость передачи теоретически должна возрас-ти 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 высоко-

скоростных 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 - бо-лее простые и "узкие" шины, сопрягаемые с основной без исполь-зования дополнительной управляющей электроники. Для

Futurebus+ такими mezzanine-bus являются Sbus и PCI.

**Шина PCI** обладает несколькими преимуществами передосновной версией VL-Bus. В соответствии со спецификацией РСI к шине могут подключаться до 10 устройств. Это, однако, не озна-чает использования такого же числа разъемов расширения - огра-ничение относится к общему числу компонентов, в том числе рас-положенных и на системной плате. Поскольку каждая плата рас-ширения РСI может разделяться между двумя периферийными устройствами, то уменьшается общее число устанавливаемых разъемов.

Шина РСI может использовать 124-контактный разъем (32-разрядная) или 188-контактный разъем (64-разрядная передача данных), при этом теоретически возможная скорость обмена со-ставляет соответственно 132 и 264 Мбайт/с. На системных платах устанавливаются обычно не более трех разъемов.

Предполагается, что стандарт PCI лучше соответствует рас-тущим потребностям в скоростной обработке данных на настоль-

325

ных машинах, поскольку превосходит стандарт VL-Bus по сложности, гибкости и функциональной насыщенности. Windows принесла в мир ПК полноцветную графику. Процессор 486 выпол-няет пересылки данных по 32-разрядной шине, тактируемой час-тотой 33 МГц. Как только выдаваемый им мощный поток графи-ческих данных попадает на шину ISA, он упирается в "узкое гор-ло". Эта шина работает на частоте всего лишь 8 МГц, а ее разряд-ность равна 16. По мере того как в прикладных программах начи-нают все шире использоваться многоцветная графика, "живое" ви-део и рендеринг трехмерных изображений, разработчикам систем и периферийных устройств пришлось предусмотреть другой спо-соб связи с узлами машины, требующими наиболее интенсивного обмена данными.

Стандартная локальная шина обеспечивает единообразный способ подключения устройств к быстродействующей шине про-цессора и тем самым позволяет устранить "узкие места " во всех новых ПК. Шина РСI поддерживает 32-разрядный канал передачи данных между процессором и периферийными устройствами, ра-ботает на высокой тактовой частоте (33 МГц) и имеет максималь-ную пропускную способность 120 Мбайт/с. Кроме того, шина PCI в некоторой степени обеспечивает обратную совместимость с су-ществующими 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 ниx - блочная передача последовательных данных. Если данные не являются последовательными, требуется дополнительное время на установку адреса каждого их элемента. Шина РСI создает между ЦП и периферийными устройствами не-который промежуточный уровень. В результате получается про-цессорно -независимая шина, как ее называет Intel. Ее легко под-ключить 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-

326

пользовать одни и те же элементы и устройства. Стандарт РС1 предусматривает обширный список дополнительных функций. К ним относится автоматическая конфигурация периферийных уст-ройств, позволяющая пользователю устанавливать новые устрой-ства без особых проблем.

РСI поддерживает целый спектр периферийных устройств и обладает средствами управления передачей данных ( что освобож-дает процессор от рутинной возни с трафиком). Нет нужды гово-рить, что все обмены по шине буферизованы. PCI легко совмести-ма с большинством известных шин. Разработаны и реализованы в виде стандартных микросхем многочисленные "мосты"; PCI/ISA, PCI/EISA, РРС/РСI и другие. Многие производители ПК практи-куют также слоты двойного назначения - например, PCI/ISA, по-зволяющие на одно и то же место устанавливать устройства ввода-вывода в различных стандартах.

**Интерфейс МПИ** с мультиплексированными линиями ад-реса и данных предназначен для обеспечения информационной и электрической совместимости устройств системы. Он реализуется на основе магистрали и логических узлов, входящих в каждое под-ключаемое к ней устройство. Устройства в совокупности состав-ляют единое адресное пространство магистрали.

* 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.
* каждый момент времени на магистрали может выполнять-ся один из трех видов взаимодействий подключенных к ней уст-ройств: передача управления магистралью, адресный обмен (оди-ночный или блочный), прерывание.

Передача управления магистралью осуществляется в соот-ветствии со схемой приоритета. Приоритет устройства определя-ется его положением на линии «разрешение на захват магистрали» (РЗМ) относительно других устройств. Приоритет устройства убывает по мере удаления устройства от микропроцессора, управ-ляющего захватом магистрали, в направлении распространения сигнала РЗМ. При процедуре передачи управления магистралью

327

активное устройство, готовое к выполнению функции ведуще-го, асинхронно выставляет запрос на захват магистрали. МП выда-ет 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 ин-терфейс может обеспечить режимы одиночного (обязательного) и блочного (необязательного) обменов данными.

Прерывание выполняемой программы МП осуществляет по запросам ВУ. При обработке запроса на прерывание процессор за-поминает состояние прерванной программы и продолжает ее после завершения прерывающей программы. Контроллер ВУ, запросив-шего прерывание, по разрешению процессора выдает вектор пре-рывания , определяющий вход в процедуру обработки программы данного прерывания. Разрешение на выдачу вектора прерывания МП выдает в соответствии с многоуровневой системой приорите-тов.

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 адре-сации ВУ, остальной для ячеек внутренних запоминающих уст-ройств.

По магистрали информация передается в двоичном позици-онном коде. Длина слова данных составляет 8 или 16 бит, а фор-мат передаваемого адреса - 16 - 24 бит.

**Интерфейс Unibus** содержит магистраль из56сигнальныхлиний. Все устройства подсоединяются к этим линиям параллель-но. Пять симплексных сигнальных линий используются для управления шиной приоритета, остальные 51 линий являются ду-плексными; 18 адресных линий используются ведущим устройст-вом для выборки ведомого устройства, с которым предстоит уста-новить связь. Одна из линий адреса задает байт, к которому при операциях с байтами происходит обращение; 16 линий данных ис-пользуются для передачи информации между ведущим и ведомым

328

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 ис-пользуются две линии синхронизации. Для передачи управления магистралью ведущему устройству используется 11 линий при-оритета (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 интерфейсаMultibus, объединяющего стандартизованные интерфейсы IEEE, VME - bus, AMS - bus и др., с сохранением состава линий и их функций.

**Интерфейсы 2 уровня** обеспечивают объединение внеш-них устройств и устройств связи с объектами (УСО), которые ис-пользуются в тех случаях , когда ВУ и УСО не имеют встроенного системного интерфейса и не могут подключаться непосредственно к системной магистрали. Наибольшее распространение здесь по-лучили интерфейс ИРПС для радиального подключения уст-ройств с последовательной передачей информации и интерфейс ИРПР для подключения устройств с параллельной передачей ин-формации. С их помощью подключаются практически все перифе-рийные устройства (дисплеи, принтеры, клавиатура, графопо-строители и т. д.), за исключением внешних запоминающих уст-ройств, предъявляющих более высокие требования к пропускной способности интерфейса.

В качестве интерфейса УСО могут быть использованы маги-страль КАМАК или специальные интерфейсные платы - контрол-леры, обеспечивающие подключение модулей УСО к системному интерфейсу. Сопряжение малого интерфейса с системной магист-ралью осуществляется при помощи контроллера К (рис. 3.18).

**Интерфейсы 3 уровня** предназначены для объединениядатчиков и исполнительных устройств. Большое разнообразие датчиков и исполнительных устройств на сегодняшний день при-вело к разработке огромного числа этих интерфейсов. Интерфейсы

329

4 уровня представляют собой интерфейсы устройств передачи данных (УПД). К ним относятся интерфейсы телеграфных, теле-фонных, высокочастотных, оптоволоконных и других каналов для передачи данных на большие расстояния. Сюда же относятся ин-терфейсы распределенных систем управления общего и специаль-ного назначения ( КАМАК МЭК - 640, МЭК - 625 - 1 последова-тельный, 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 специально-го 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 пе-редатчик и приемник, но одновременная передача в двух направ-лениях невозможна. Для изменения направления передачи требу-ется 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 асин-хронного (обмен с квитированием) принципов обмена. В первом случае устройство - источник (контроллер) определяет темп выда-чи и приема информации и синхронизирует все процессы, связан-ные с трансляцией данных . Обычно синхронизируется прохожде-ние в линии каждого бита, группы битов (символа) и сообщения.

Асинхронный принцип передачи в интерфейсах, как правило, ocнован 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.

330

Приемник фиксирует поступление сигнала готовности источ-ника, 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 асин-хронной более эффективно используется канал связи и достигается лучшая помехозащищенность передаваемых данных, в интерфей-сах автоматизированных систем научных исследований применя-ют, как правило, асинхронный способ передачи. Это обусловлено возможностью передавать в асинхронном режиме данные со ско-ростью, соответствующей быстродействию того устройства, с ко-торым в данный момент времени происходит обмен информацией (автоматическая подстройка скорости передачи данных).

**Интерфейс AGP** предназначен для вывода информации навнешние устройства, в том числе отображения данных. Она со-держит шину и устройство передачи информации (видеоускори-тель), образующие интерфейсную схему . В настоящее время наи-большее применение получил интерфейс AGP

В начале 1997 г. фирмой Intel был разработан новый стан-дарт для вывода графики, получивший название AGP (Accelerated Grafics 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 до-бавки 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 дос-тигает значения в 532 Мбайт/с.

331

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 распреде-ляются между основной и локальной памятью, что, в свою оче-редь, влияет на качество отображаемой картинки и частоту смены кадров.

* режиме DMA для графики используется только локальная память видеоускорителя, а данные, расположенные вне ее, предва-рительно загружаются в локальную память и лишь затем обраба-тываются видеопроцессором. При этом AGP выполняет роль бы-строй шины.
* режиме DIME для построения изображения видеоускори-тель 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 сис-темной памяти в локальную, а интерпретируются «на месте».

Необходимо отметить одну важную особенностью AGP, ко-торая состоит в том, что память под текстуры выделяется опера-ционной системой по требованию исполняемой программы и оста-ется доступной для него. Поскольку текстура может занимать бо-лее одной страницы оперативной памяти (более 4 Кбайт), то в спе-цификацию AGP включена таблица переадресации графики, со-держимое которой должно быть согласовано с таблицами переад-ресации операционной системы. Таким образом, поддержка AGP не может ограничиваться драйвером производителя, здесь требу-ется поддержка на уровне операционной системы. Такая поддерж-ка реализована, например, в версии Windows 98.

Преимуществом при использовании AGP является следую-

щее:

- интерфейс реализован по принципу «соединение точка-точка», при котором отсутствуют проблемы с арбитражем шин,

- в интерфейсе предусмотрены раздельные шины для пере-дачи команд и данных, повышающие пропускную способность ин-терфейса,

- применение конвейеризации и технологии отложенного выполнения команд, позволяющие вплотную приблизиться к тео-ретическому скоростному пределу интерфейса.

332

Принимая во внимание перечисленные преимущества ин-терфейса AGP разработано большое количество видеоускорите-лей, совместимых со спецификацией AGP, среди которых можно отметить Asus 3DexPlorer 3000, ATI Xpert@Work, Diamond Vi-per330, 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 видеопро-граммы используют существенно разные функции. Большое раз-нообразие функций является единственной главной причиной, не позволяющей однозначно определить быстродействие. Однако проведенные приближенные оценки показывают, что для боль-шинства программ трехмерной графики быстродействие микро-процессорных систем вывода видеоинформации с использованием видеоускорителей , использующих шину PCI, и систем с использо-вание интерфейса AGP приблизительно одинаково.

**12. Организация памяти МПС**

Память современных вычислительных систем имеет иерар-хическую многоуровневую структуру. Чем выше уровень, тем выше требуемое быстродействие соответствующей памяти. Ос-новная или оперативная память относится к верхнему уровню па-мяти.

Сравнительно небольшая емкость ОП (до нескольких десят-ков Мбайт) компенсируется практически неограниченной емко-стью внешний запоминающих устройств на магнитных дисках (до нескольких Гбайт). Однако эти устройства сравнительно медлен-ные, и время обращения за данными для дисков составляет десят-ки миллисекунд. Поэтому вычислительный процесс должен проте-кать с возможно меньшим числом обращений к внешним запоми-нающим 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-

333

цессора и ОП. Возникающая проблема выравнивания их пропу - скных способностей решается путем использования буферных па-мятей небольшой емкости и повышенного быстродействия. Они используются для хранения команд и данных, относящихся к об-рабатываемому участку программы.

Оперативная память является наиболее дефицитным ресур-сом в вычислительных системах, которым надо пользоваться эко-номно и эффективно. Проблема усложняется при переходе к муль-типрограммным системам, так как в них ОП одновременно ис-пользует несколько программ (заданий). В таких системах важным стоит вопрос исключения несанкционированного воздействия од-них программ на другие. Это достигается с помощью механизма защиты памяти.

**Защита памяти.** Если в памяти одновременно могут нахо-диться несколько независимых программ, необходимы специаль-ные меры по предотвращению или ограничению обращений одной программы к областям памяти, используемым другими програм-мами. 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 опе-рационной системы. Другими словами, надо исключить воздей-ствие программы пользователя на работу программ других поль-зователей и программ операционной системы.

Чтобы воспрепятствовать разрушению одних программ дру-гими, 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 допуска-ется обращение других программ к этой области памяти для счи-тывания данных.

В других случаях, например при ограничениях на доступ к информации, хранящейся в системе, необходимо иметь воз-можность запрещать другим программам производить как запись, так и считывание в данной области памяти. Такая защита от запи-си и считывания помогает отладке программы, при этом осущест-вляется контроль каждого случая выхода за область памяти своей программы.

334

Для облегчения отладки программ желательно выявлять и такие характерные ошибки в программах , как попытки использо-вания данных вместо команд или команд вместо данных в собст-венной программе, хотя эти ошибки могут и не разрушать инфор-мацию.

Необходимо отметить следующие варианты дифференциро-ванной защиты при различных операциях с памятью:

задается отношение к области памяти чужой программы, определяющее, относится защита памяти только к операции запи-си 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, взято-му только из счетчика команд;

г) разрешается обращение по адресу из любого регистра, кро-ме счетчика команд.

Если нарушается защита памяти, исполнение программы приостанавливается и вырабатывается запрос прерывания по на-рушению защиты памяти.

3ащита от вторжения программ в чужие области памяти мо-жет быть организована различным образом, при этом реализация защиты не должна заметно снижать производительность МПС и требовать слишком больших аппаратурных затрат.

Защита отдельных ячеек памяти. В управляющих вычисли-тельных комплексах необходимо обеспечить возможность отладки новых программ параллельно с функционированием находящихся

* памяти рабочих программ, управляющих технологическим про-цессом. Это может быть достигнуто выделением в каждой ячейке памяти специального «разряда защиты». Установка «1» в этот раз-ряд запрещает производить запись в данную ячейку.

В системах с мультипрограммной обработкой большого чис-ла программ защищаются не отдельные ячейки, а области памяти или блоки, на которые делится память, при этом часто предусмат-ривается возможность указывать для разных программ различные

335

допустимые режимы обращения к отдельным областям или блокам памяти.

Существует несколько методов защиты памяти, среди кото-рых можно выделить метод граничных регистров и метод ключей защиты.

Метод граничных регистров состоит во введении двух гра-ничных регистров, указывающих верхнюю и нижнюю границы области памяти, куда программа имеет право доступа. При каждом обращении к памяти проверяется, находится ли используемый ад-рес в установленных границах; при выходе за границы обращение к памяти подавляется и формируется запрос прерывания, пере-дающий управление операционной системе.

Содержание граничных регистров устанавливаются операци-онной системой перед тем, как для очередной исполняемой про-граммы начнется активный цикл. Если для динамичного распреде-ления памяти используется базовый регистр, то он одновременно определяет и нижнюю границу. Верхняя граница подсчитывается операционной системой в соответствии с длиной программы опе-ративной 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, назы-ваемый ключом защиты памяти, а каждой программе, принимаю-щей 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 специаль-ной памяти защиты ключей, более быстродействующей, чем ОП.

**Пропускная способность процессора и МПС. Kэш – память.** Непрерывный рост производительности(скорости рабо-ты) МПС проявляется, в первую очередь, в повышении скорости работы электронных схем, а также специальных архитектурных решений — конвейерная и векторная обработка данных, кэширо-вание памяти и др.

336

Быстродействие оперативной памяти также растет, но все время отстает от быстродействия аппаратурных средств про-цессора, в значительной степени потому, что одновременно про-исходит 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 ука-занное противоречие и согласовать пропускные способности па-мяти и процессора помогают специальные структурные решения.

Конвейеризация процедур цикла выполнения команды (ра-бочего цикла машины) в простейшем случае предполагает выпол-нение параллельно во времени операции в АЛУ с выборкой из па-мяти следующей команды.

Буферизация — использование включенных между процес-сором и ОП существенно более, чем ОП, быстродействующих бу-ферных памятей сравнительно небольшой емкости. На рис. 76 по-казана 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 - памятей. 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 не-сколько команд. Обмен информацией между кэш-памятью и мик-ропроцессором осуществляется высокоскоростной локальной ши-ной, при этом кэш-команд имеет свою шину, а кэш-данных свою. В последних разработках (Pentium III и IV) работа шин осуществ-ляется 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 высо-коскоростным буфером между процессором и относительно мед-ленной основной памятью. При некоторых обращениях к памяти

337

соответствующие значения заносятся в кэш, и в ходе следую-щих операций чтения по тем же адресам обращения происходят только к кэш - памяти.

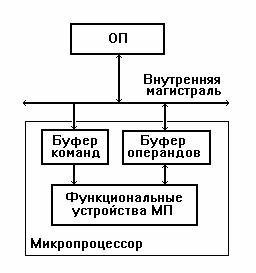


Рис. 76 Для обращения к кэш - памяти, размещенной вне кристалла

процессора, но на одной плате с процессорной БИС, может потре-боваться несколько циклов памяти, тогда как при обращении к кэш - памяти, находящейся в процессорном кристалле , может ока-заться достаточно одного такого цикла; однако даже размещение кэш - памяти на одной плате с процессором позволяет избежать большого числа циклов ожидания, которые неизбежны при ра-боте с памятью, расположенной на отдельной плате и взаимодей-ствующей 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. Выбор-ка ее содержимого может производиться произвольным образом, и, следовательно, в адресном диапазоне кэш-памяти можно раз-местить командные циклы с входящими в них командами перехо-дов.

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

338

информация окажется в кэш -памяти. Если при объеме кэш 4 Кбайт и длине строки 4 байт вероятность удачных обращений со-ставляет 80%, то при увеличении длины строки до 8 байт эта ве-личина может достигнуть 85%. Однако дальнейшее увеличение длины строки не приводит к заметному росту вероятности удач-ных обращений . Так, например, увеличение длины строки еще в два раза (до 16 байт) позволяет довести вероятность удачных об-ращения примерно до 87%.

Полная производительность памяти является функцией вре-

|  |  |
| --- | --- |
| мени доступа к | кэш-памяти, вероятности удачных обращений к |
| кэш и временем | обращения к основной памяти, которое проис- |

ходит при неудачном обращении к кэш. Возможна, например, си-туация , 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 со-

ставляет (0,8 × 120) + 0,2 × (600 + 120) = 240 нс.

При обращении процессора к ОП для считывания в кэш пе-редается блок информации, содержащий нужное слово. При этом происходит опережающая выборка, так как высока вероятность того, что ближайшие обращения будут происходить к словам это-го же блока, уже находящимся в кэш-памяти. Это приводит к зна-чительному уменьшению среднего времени, затрачиваемого на выборку данных.

Эффективность кэш-памяти, зависящая от ее емкости, разме-ра слова, соотношения времен считывания слова из кэш-памяти и блока из ОП проявляется в уменьшении среднего времени, затра-чиваемого на выборку слова данных.

Можно выделить два типа кэш - памяти:

с запоминанием новой информации одновременно в кэш и

* ОП («сквозное запоминание»), при этом в ОП есть всегда по-следняя копия хранящейся в кэш информации. Однако в этом слу-чае длинный цикл ОП снижает производительность процессора;

с запоминанием новой информации только в кэш и копи-рованием ее в ОП только при передаче в другие устройства или при вытеснении из кэш-памяти.

339

**Динамическое распределение памяти. Виртуальная страничная и сегментная память.** Эффективное распределениересурсов памяти между программами не может быть статиче-ским, т. е. не может производиться предварительно до пуска про-граммы . В процессе обработки программ потребности в ресурсах памяти отдельных программ изменяются, что заранее не может быть учтено. Зачастую целесообразнее распределять память меж-ду программами динамически непосредственно в ходе вычисли-тельного процесса, т. е. осуществлять динамическое распределе-ние памяти. При этом должна обеспечиваться возможность неза-висимой работы программистов над своими программами, подле-жащими мультипрограммной обработке.

Необходимо отметить , что динамическое распределение па-мяти не должно приводить к дроблению ее свободного простран-ства — фрагментации памяти, затрудняющему ее использование. Это достигается организацией одноуровневой виртуальной памя-ти, допускающей адресацию на все адресное пространство. Размер его определяется количеством разрядов, которые могут быть ис-пользованы для представления адреса. К тому же в мультипро-граммных системах размещение всех исполняемых программ пол-ностью в ОП во многих случаях невыполнимо: программы часто имеют большую длину, а емкости существующих ОП ограничен-ны. Однако нет принципиальной необходимости в том, чтобы вся программа находилась в ОП, так как в любой момент времени ра-бота программы концентрируется на определенных сравнительно небольших участках. Таким образом, в ОП следует хранить только используемые в данный период времени части программ, а неис-пользуемые части могут располагаться во ВЗУ. Программируя свою программу, программист не знает, в комбинации с какими программами будет выполняться его программа, какое место в па-мяти 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.

340

Осуществление динамического распределения чисто про-граммным путем привело бы к значительным потерям машинного времени. Целесообразнее пользоваться для этой цели аппара-турными средствами.

Один из способов динамического распределения памяти ос-нован на использовании базовых регистров. Операционная систе-ма каждой исполняемой программе ставит в соответствие свой ба-зовый адрес. Базовые адреса программ находятся в общих регист-рах. При выполнении программы реальный или физический адрес образуется суммированием базового и относительного адресов. При динамическом распределении памяти с помощью базовых ре-гистров программа ( или, по крайней мере, та ее часть, адрес кото-рой 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, мо-жет 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 по-надобиться сдвиг содержимого памяти. На рис. 77, а показано рас-пределение памяти между программами А, В, С, D, из которых две, например, А и D,. являются в данный момент наименее актив-ными и следовательно, могут рассматриваться как кандидаты на удаление во внешнюю память

Если вновь вводимая программа Е (рис. 77, б) больше лю-бой из программ А и D, то для ее размещения в памяти необходи-мо, как показано на рис. 77, в, сдвигать программы В и С. Это пе-ремещение связано с потерей времени. Более того, в ряде систем подобное перемещение требует выполнения заново операции ре-дактирования связей в программе и новой заг-рузки программы, что в ряде случаев ведет к существенному снижению быстродей-ствия системы в целом. Эти и некоторые другие недостатки в рас-пределении памяти отсутствуют в виртуальной памяти со стра-ничной организацией.

341



Рис.77

Виртуальная память есть способ, организации памяти муль-типрограммной вычислительной системы , при котором достигает-ся 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 опера-ционной системы и процессов их функционирования.

Принцип виртуальной памяти предполагает, что пользова-тель при подготовке своей программы имеет дело не с физической ОП, действительно работающей в составе вычислительной систе-мы и имеющей некоторую фиксированную емкость, а с виртуаль-ной (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 про-странство системы независимо от объема ее физической памяти и объемов памятей, необходимых для других программ , участвую-щих 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 ре-альные адреса действующей памяти (в так называемые физиче-ские адреса.). Преобразование виртуальных адресов в физические упрощается и устраняется фрагментация памяти, если физическую и виртуальную память разбить на блоки двух типов. К первому

342

типу относятся блоки, образующие страницы, ко второму типу - сегменты.

При страничной организации память разбивается на страни-цы фиксированного размера, например, по 512 байт. В них осуще-ствляется загрузка программ под управлением операционной сис-темы. Страницам виртуальной и физической памяти присваивают номepa, называемые номерами соответственно виртуальных и фи-зических 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 па-мяти дает определенные преимущества. Когда новая программа загружается в ОП, она может быть направлена в любые свободные

* данный момент физические страницы независимо от того, распо-ложены они подряд или нет. Не требуется перемещения информа-ции 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 загру-жается начальная страница программы и ей передается управле-ние. Если по ходу работы делается попытка выборки слов из дру-гой страницы, то производится автоматическое обращение к опе-рационной системе, которая осуществляет загрузку из ВЗУ тре-буемой страницы.

На рис. 78 показано соответствие между виртуальной и фи-зической памятями, устанавливаемое страничной таблицей. Оче-видно, 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 пе-рерабатывается ею каждый раз, когда в распределении памяти производятся изменения. Процедура обращения к памяти состоит

* 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,

343

представляет собой физический адрес, по которому происходит обращение к ОП.

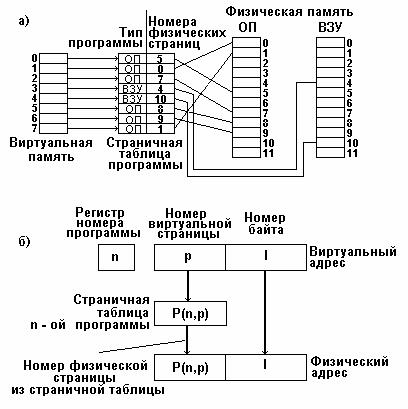


Рис.78

Если страничная таблица указывает на размещение требуе-мой информации во внешней памяти, то обращение к ОП не может состояться немедленно: операционная система должна организо-вать передачу из внешней памяти в ОП нужной страницы.

Для каждой из программ , обрабатываемых в мультипро-граммном режиме, организуется своя область виртуальной памяти

344

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 не-большая сверхоперативная память, куда передается из ОП стра-ничная таблица активной программы. В другом варианте в сверх-оперативной памяти могут находиться сведения о номерах вирту-альных 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 ассоциа-тивная 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 сплош-ную область памяти, тогда как при страничной организации мо-дуль разбивается на страницы.

Логический адрес, вырабатываемый процессором для систе-мы сегментной виртуальной памяти, состоит из номера сегмента и смещения. Таблица сегментов содержит значения базовых адресов (представляющих собой начальные адреса сегментов в физиче-ской памяти) и границ (определяющих объемы сегментов ). Вели-чина смещения не должна превышать значение границы. Защита сегмента, содержащего законченный программный модуль, обес-печивается путем однократного задания прав на использование и доступ для записи и чтения.

345

В процессе работы системы с сегментной виртуальной памятью в ней могут появляться так называемые «дырки», пред-ставляющие собой неиспользуемые области, которые требуют объединения. В системах реального времени это действие может служить причиной определенных неудобств. Сегментная вирту-альная память должна быть снабжена алгоритмом размещения, с помощью которого осуществляется поиск неиспользуемых зон памяти для размещения каждого сегмента. В системах странич-ной виртуальной памяти необходимости в этих сложных алго-ритмах нет; кроме того, при страничной организации виртуальной памяти отсутствует опасность появления «дырок» между про-граммами (внешней фрагментации), а величина «дырок» внутри программы (внутренней фрагментации) не может превышать объема одного блока для каждой программы.

**Сегментно - страничная организация памяти.** До сихпредполагалось, что виртуальная память, которой располагает программист, представляет собой непрерывный массив с единой нумерацией байт. Однако программа обычно состоит из несколь-ких массивов — подпрограмм, одной или нескольких секций дан-ных. Так как заранее длины этих массивов неизвестны, то удобно, чтобы при программировании каждый массив имел свою собст-венную нумерацию байт, начинающуюся с нуля и про-должающуюся в возрастающем порядке. Желательно также, чтобы составленная таким образом программа могла работать при дина-мическом распределении памяти , не требуя от программиста уси-лий по объединению различных ее частей в единый массив. Эта задача решается в некоторых вычислительных системах путем ис-пользования 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 дополни-тельные разряды левее номере страницы; эти разряды определяют номер сегмента.

Возникает определенная иерархия в организации программ, состоящая из четырех ступеней: программа → сегмент → страница

346

→ байт. Этой иерархии программ соответствует иерархия таб-лиц, служащих для перевода виртуальных адресов в физические. Программная таблица для каждой программы, загруженной в сис-тему, указывает начальный адрес соответствующей сегментной таблицы. Сегментная таблица перечисляет сегменты данной про-граммы с указанием начального адреса страничной таблицы, отно-сящейся к данному сегменту. Страничная таблица определяет рас-положение каждой из страниц сегмента в памяти. Страницы сег-мента могут располагаться не подряд, часть страниц данного сег-мента может находиться в оперативной памяти, остальные — во внешней.

Рассмотрим для примера организацию сегментно-страничной виртуальной памяти некоторой вычислительной сис-темы. 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 вирту-альный адрес (как и физический) будет иметь длину 24 разряда, причем поле номера сегмента будет занимать 8 или 4 старших раз-ряда 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 про-граммной таблице нет необходимости, так как для каждой актив-ной 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, установлен-ный в управляющем регистре 1, суммируется с номером сегмента из виртуального адреса. В результате образуется адрес, по которо-

347

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 но-мер, 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 од-ноименную процедуру, называемую прерыванием по странично-му 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 формиру-ется физический адрес.

Далее выполняется запрошенное программой обращение к ОП. Одновременно информация о текущей странице (номерах про-граммы, сегмента, виртуальной и соответствующей физической страницы) помещается в сверхоперативную ассоциативную память или в блок быстрой переадресации небольшой емкости.

**Ассоциативная память.** Ассоциативная память хранитуказанные данные для небольшого числа недавно использовав-шихся страниц. При наличии ассоциативной памяти значительно ускоряется процесс преобразования адресов, так как на каждом участке вычислительного процесса обращения к ОП сосредотачи-ваются 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 адре-сов без дополнительных обращений к ОП.

348

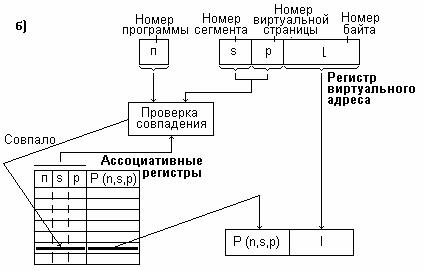
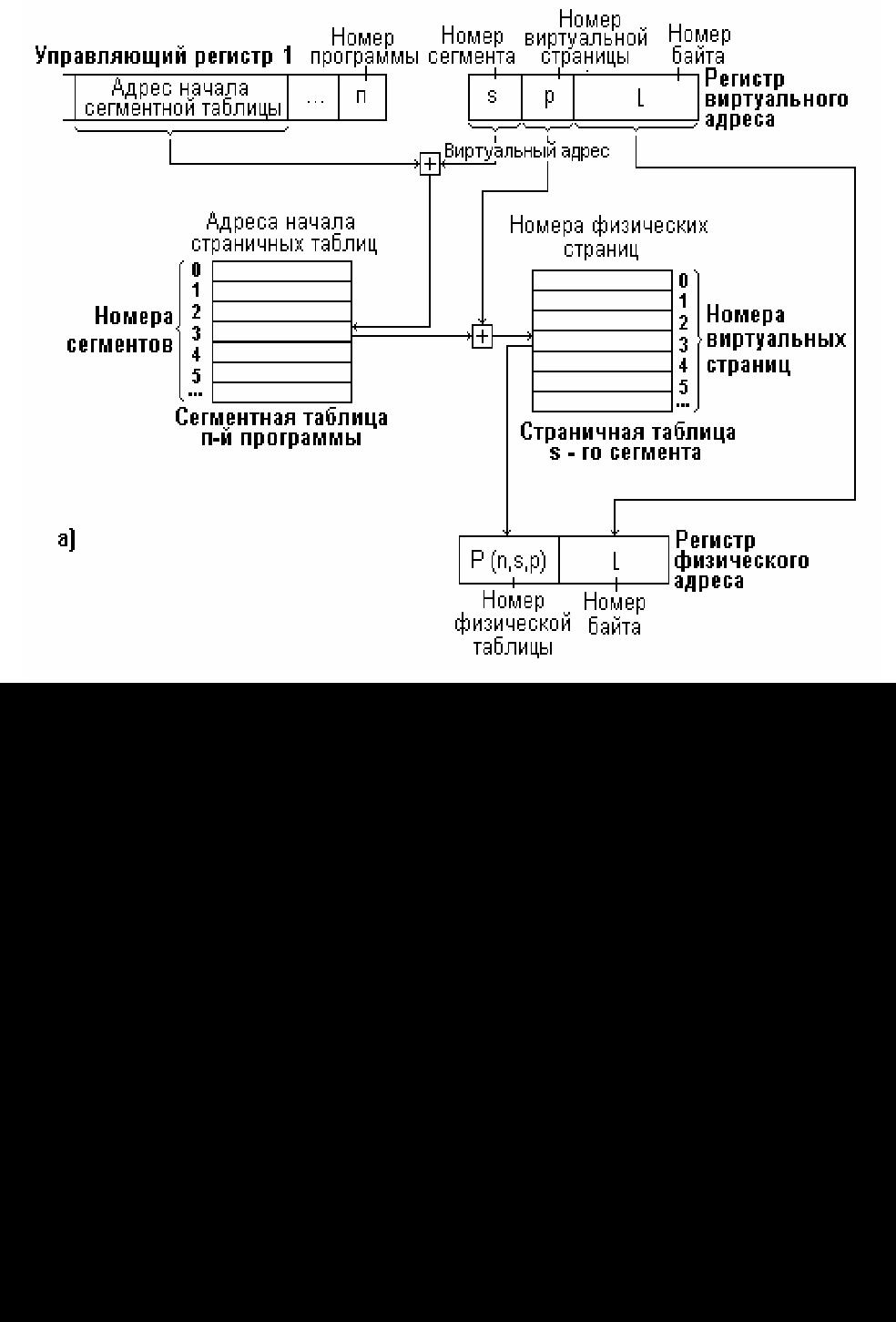


Рис.79.

349

Преобразование адресов всегда начинается с просмотра ассоциативной памяти. Если оказывается, что в одной из ее строк (ассоциативном регистре) хранится информация о странице, к ко-торой должно произойти обращение, то из этой строки непосред-ственно выбирается номер физической страницы и дополнитель-ные обращения к ОП ( к сегментной и страничной таблицам) не производятся (рис .79,6). Если нужной информации нет в ассоциа-тивной памяти, то делается попытка сократить время преобразова-ния путем исключения одного дополнительного обращения к ОП (первый этап на рис. 79, а). Может оказаться так, что страница, к которой происходит обращение, принадлежит сегменту предыду-щего 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 преды-дущего обращений, первый этап преобразования исключается, ис-пользуется сохраненный адрес начала сегментной таблицы и вы-полняется только второй этап преобразования, т. е. производится только одно дополнительное обращение к ОП. Если номера сег-ментов не совпадут, то реализуется полная процедура преобразо-вания адресов, показанная на рис. 79, а.

Дополнительные обращения к ОП сопровождаются занесе-нием информации о текущей странице в ассоциативную память. Если в ассоциативной памяти не оказывается свободного регистра (строки), то данные о новой странице записываются на место дан-ных, которые дольше других не использовались в процессе преоб-разования адресов.

**13. Технологические аспекты полупроводниковой технологии**

Совершенствование технологического процесса изготовле-ния микропроцессоров – это главный атрибут повышения их бы-стродействия и надежности. Переход на новые техпроцессы явля-ется очевидным шагом, но технологам это дается каждый раз все с большим трудом. Современные процессоры выполняются по тех-нологии 0,13 и 0,09 мкм, причем последняя была введена в 2004 году. Как видно, для этих техпроцессов соблюдается закон Мура,

350

который гласит, что каждые два года частота кристаллов удваи-вается 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 SiO2, изолятора, находя-щегося между затвором и каналом, и выполняющего вполне по-нятную функцию - барьера для электронов, предотвращающего утечку тока затвора. Очевидно, что чем толще этот слой, тем луч-ше он выполняет свои изоляционные функции, но он является со-ставной частью канала, и не менее очевидно, что если мы собира-емся 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 SiO2, 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, подни-мать рабочий ток, со всеми вытекающими условиями.

Изготовление микропроцессора - это сложнейший процесс, включающий более 300 этапов. Микропроцессоры формируются на поверхности тонких круговых пластин кремния - подложках, в результате определенной последовательности различных процес-сов обработки с использованием химических препаратов, газов и ультрафиолетового излучения.

351



Рис. 80

Подложки обычно имеют диаметр 200 миллиметров, или 8 дюймов. Однако корпорация Intel уже перешла на пластины диа-метром 300 мм, или 12 дюймов. Новые пластины позволяют полу-чить почти в 4 раза больше кристаллов, и выход годных значи-тельно выше. Пластины изготавливают из кремния, который очи-щают, плавят и выращивают из него длинные цилиндрические кристаллы. Затем кристаллы разрезают на тонкие пластины и по-лируют их до тех пор, пока их поверхности не станут зеркально гладкими и свободными от дефектов. Далее последовательно цик-лически повторяясь производят термическое оксидирование (фор-мирование пленки SiO2), 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 совер-шить более 300 операций. Весь процесс производства процессоров можно разделить на несколько этапов: выращивание диоксида кремния и создание проводящих областей, тестирование, изготов-ление корпуса и доставка.

352

Процесс производства микропроцессора начинается с "выращивания" на поверхности отполированной пластины изоля-ционного слоя диоксида кремния. Осуществляется этот этап в электрической печи при очень высокой температуре. Толщина ок-сидного 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 материа-ла – фоторезист, на который с помощью ультрафиолетового излу-чения проецируют изображение прозрачных участков шаблона, или фотомаски. Маски изготавливают при проектировании про-цессора и используют для формирования рисунков схем в каждом слое процессора. Под воздействием излучения засвеченные участ-ки 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 фо-тослой, в результате чего на полупроводниковой пластине остает-ся рисунок из диоксида кремния. В результате ряда дополнитель-ных операций фотолитографии и травления на пластину наносят также поликристаллический кремний, обладающий свойствами проводника. В ходе следующей операции, называемой "легирова-нием", 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 соеди-нений 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 процес-се и процессах предыдущих поколений Intel применяла алюминий. И медь, и алюминий - отличные проводники электричества. При использовании 0,18-мкм техпроцесса использовалось 6 слоев, при

353

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 несколь-ких недель.

Чтобы выдержать воздействия , которым подвергаются под-ложки в процессе нанесения слоев, кремниевые пластины изна-чально 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 "по-худевшей" пластины наносят слой специального материала, кото-рый улучшает последующее крепление кристалла к корпусу. Кро-ме того, этот слой обеспечивает электрический контакт между задней поверхностью интегральной схемы и корпусом после сбор-ки.

После этого пластины тестируют, чтобы проверить качест-во выполнения всех операций обработки. Чтобы определить, пра-вильно ли работают процессоры, проверяют их отдельные компо-ненты. Если обнаруживаются неисправности, данные о них анали-зируют, 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 изготов-ленных процессоров заданным требованиям.

После тестирования пластины отправляются в сборочное производство, где их разрезают на маленькие прямоугольники, каждый из которых содержит интегральную схему. Для разделе-ния пластины используют специальную прецизионную пилу. Не-работающие кристаллы отбраковываются.

Затем каждый кристалл помещают в индивидуальный кор-пус. Корпус защищает кристалл от внешних воздействий и обес-печивает его электрическое соединение с платой, на которую он будет впоследствии установлен. Крошечные шарики припоя, рас-положенные в определенных точках кристалла, припаивают к

354

электрическим выводам корпуса. Теперь электрические сигна-лы могут поступать с платы на кристалл и обратно.

В будущих процессорах компания Intel применит техноло-гию BBUL, которая позволит создавать принципиально новые кор-пуса с меньшим тепловыделением и емкостью между ножками

CPU.

После установки кристалла в корпус процессор снова тес-тируют, чтобы определить, работоспособен ли он. Неисправные процессоры отбраковывают, а исправные подвергают нагрузочным испытаниям: воздействию различных температурных и влажност-ных режимов, а также электростатических разрядов. После каждо-го 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 подни-мать частоты процессоров также безболезненно. В конце 2003 года специалисты Intel сделали очень важное заявление о разработке новых материалов для полупроводниковых транзисторов будуще-го. Прежде всего , речь идет о новом диэлектрике затвора транзи-стора с высокой диэлектрической проницаемостью (так называе-мый «high-k»-материал), который будет применяться взамен ис-пользуемого сегодня диоксида кремния SiO2 (рис. 81), а также о новых металлических сплавах, совместимых с новым диэлектри-ком затвора. Решение, предложенное исследователями , снижает ток утечки в 100 раз, что позволяет вплотную подойти к внедре-нию производственного процесса с проектной нормой 45 наномет-ров. Оно рассматривается экспертами как маленькая революция в мире микроэлектронных технологий.

Чтобы понять, о чем идет речь, взглянем сначала на обыч-ный МОП-транзистор, на базе которого делаются МП. В нем за-твор 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).

355

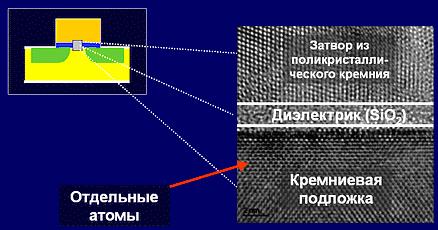


Рис. 81

Столь малая толщина диэлектрика необходима для получе-ния не только малых габаритов транзистора в целом , но и для его высочайшего быстродействия (заряженные частицы передвигают-ся быстрее через затвор, в результате чего такой VT может пере-ключаться до 10 миллиардов раз в секунду ). Упрощенно - чем ближе затвор к каналу транзистора (то есть, чем тоньше диэлек-трик), тем «большее влияние» в плане быстродействия он будет оказывать на электроны и дырки в канале транзистора.

Но с другой стороны, такой тонкий диэлектрик пропускает б***о***льшие 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, препятствую-щих дальнейшему наращиванию количества транзисторов на кри-сталле. Более того, чем меньше по размерам мы делаем транзи-стор, тем тоньше нужно делать подзатворный диэлектрик. Но при его толщинах менее 1 нм резко (по экспоненте) возрастают тун-нельные токи утечки, что делает принципиально невозможным создание традиционных транзисторов менее определенных «гори-зонтальных» размеров (если при этом мы хотим получить от них хорошие скоростные характеристики). По оценкам экспертов, в

356

современных чипах почти 40% энергии может теряться из-за утечек.

Поэтому важность открытия ученых Intel нельзя недооце-нивать. После пяти лет исследований в лабораториях корпорации разработали специальный материал, позволяющий заменить тра-диционный диоксид кремния в обычном маршруте производства микросхем. Требования к такому материалу весьма серьезны: вы-сокая химическая и механическая (на атомарном уровне) совмес-тимость с кремнием, удобство производства в едином цикле тра-диционного кремниевого техпроцесса, но главное - низкие утечки и высокая диэлектрическая проницаемость.

Если мы боремся с утечками, то толщину диэлектрика нуж-но 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 диэлектри-ка. Если проницаемость объемного диоксида кремния равна 4 (или чуть меньше в сверхтонких слоях), то разумной величиной ди-электрической проницаемости нового «интеловского» диэлектрика можно считать величину в районе 10-12. Несмотря на то, что ма-териалов с такой диэлектрической проницаемостью немало (кон-

357

денсаторные керамики или монокристалл кремния), тут не ме-нее важны факторы технологической совместимости материалов. Поэтому для нового high-k-материала был разработан свой высо-коточный процесс нанесения, во время которого формируется один молекулярный слой этого материала за один цикл (рис. 83). Исходя из этой картинки можно предположить, что новый ма-териал - это тоже оксид. Причем монооксид, что означает приме-нение материалов преимущественно второй группы, например,

магния, цинка или даже меди.

Но диэлектриком дело не ограничилось. Потребовалось сменить и материал самого затвора - привычный поликристалли-ческого кремния. Дело в том, что замена диоксида кремния на high-k-диэлектрик ведет к проблемам взаимодействия с поликри-сталлическим кремнием (ширина запрещенной зоны транзистора определяет минимально возможные для него апряжения).

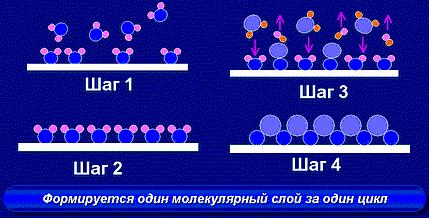


Рис. 83

Эти проблемы удается устранить, если использовать специаль-ные металлы для затворов транзисторов обоих типов (n-МОП и p-МОП) в сочетании с особым технологическим процессом. Благо-даря этой комбинации материалов удается достичь рекордной производительности транзисторов и уникально низких токов утеч-ки, в 100 раз меньших, чем при использовании нынешних мате-риалов (рис. 84).

358



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 ис-пользуется в 90-нанометровых процессорах Prescott и Dothan. На-конец-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).

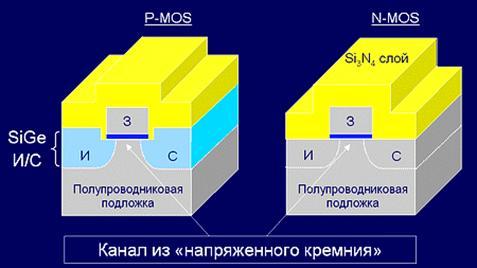


Рис. 85

359

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 крем-ния (Si3N4), который за счет механических напряжений немного (на доли процента) растягивает (в направлении протекания тока) кристаллическую решетку кремния под затвором, в результате че-го рабочий ток канала возрастает на 10% (условно говоря, элек-тронам становится более просторно двигаться в направлении ка-нала). 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 возрас-тает 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 ис-пользуется 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 - пер-вая и пока единственная компания, которая использует этот low-k-материал для изоляции межсоединений.

360

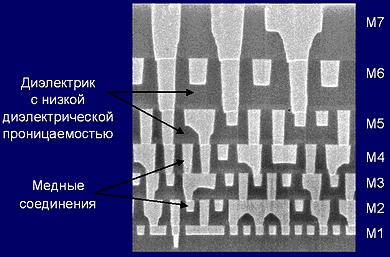


Рис. 86

**Приложение 1. История развития компьютеров – период до появления первого ПК**

* 1617 год. Джон Непер (John Napier) создал деревянную ма-шину для выполнения простейших вычислений.
* 1642 год. Блез Паскаль (Blaise Pascal) описал машину для суммирования чисел.
* 1822 год. Чарльз Баббадж (Charles Babbage) представил ме-ханическое устройство, названное позднее аналитической маши-ной, которую можно считать первой настоящей вычислительной машинной.
* 1906 год. Ли Ди Форест (Lee DeForest) запатентовал ваку-умный триод, использовавшийся в качестве переключателя в пер-вых электронных компьютерах.
* 1945 год. Джон фон Нейман (John Von Newmann) написал статью First Draft of a Report on the EDVAC, в которой была опи-

сана архитектура современных программируемых компьютеров.

* 1946 год. Джоном Мошли (John Mauchly) и Дж. Преспером Экертом (J. Presper Eckert) создана электронно-вычислительная машина ENIAC.
* 1947год. 23 декабря Джоном Бардином (John Bardeen), Уол-тером Браттейном (Walter Brattian) и Уильямом Шокли (William Shockley) был успешно протестирован первый транзистор, совер-шивший переворот в полупроводниковой технике.

361

* 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 Bereau был установлен компьютер

UNIVAC I.

* 1953 год. Фирма IBM создала первый электронный компь-ютер 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 кальку-лятор 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 начинает-ся 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 пла-стине.
* 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 помо-щью 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 фир-мы IBM занимала 81,2% компьютерного рынка; в этом году IBM анонсирована серию систем 1400.

362

* 1964 год. Суперкомпьютер CDC 6600, созданный Сей-муром 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 ре-альном времени на системе IBM SABRE.
* 1965 год. Фирма Digital Equipment Corporation анонсирова-

ла первый успешный коммерческий проект мини-компьютера

PDP-8.

* 1966 год. Фирма Hewlett Packard представила компьютер для бизнеса НР-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 НР-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.

363

• 1974 год. Компания Micro Instrumentation Telemetry Systems (MITS), занимающаяся электроникой в городе Альбукерке (шт. Нью-Мексико) объявила о разработке небольшого компьюте-ра для индивидуального пользования. Эд Робертс и двое его парт-неров создали небольшой сборный компьютер. Он получил назва-

ние Altair.



• 1974 год. Scelbi (SCientific ELectornic and Biological) Com-puter Consulting представила машину на базе процессора Intel — 8008. Она имела 1 кбайт программируемой памяти и была предна-значена в основном для научного применения.



364

* 1974 год. В исследовательском центре Пало Альто фир-мы Xerox 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 процес-

сор Cray I

* 1977 год. Фирма Tandy Radio Snack выпустила компьютер

TSR-80.

* 1977 год. Создан компьютер Apple II.

365



* 1977 год. Фирма Commodore выпустила компьютер РЕТ

(Personal Electronic Transactor).

* 1978 год. Фирмой Digital Equipment Corporation создан ком-

пьютер 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 компьютер-ного "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 ко-торого 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 персональ-ный компьютер РС.

366

* 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 РС.



• 1984 год. Фирма Apple стала выпускать первый самый ус-пешный компьютер с графическим интерфейсом пользователя, ко-торый принес 1,5 млн долларов только за этот год.

367



• 1984 год. Фирма IBM создала компьютер РС-АТ 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 компью-теров семейства PS/2, в которых был установлен 3,5-дюймовый дисковод и VGA-видеоадаптер.
* 1988 год. Один из основателей Apple Стив Джобе (Steve Jobs) покидает эту фирму и создает собственную компанию NeXT.
* 1988 год. Фирма Compaq и другие производители РС - со-вместимых систем разработали новую, улучшенную архитектуру компьютера.

368

* 1988 год. Роберт Моррис (Robert Morris) создает и запус-кает своего "червя" в сеть ARPAnet; заражено по различным оцен-кам 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 раз-работан 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 системуWindows 95.
* 1997 год. Фирма Intel выпустила процессор Pentium II, по-строенный 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 - бо-лее дешевую версию Pentium II.
* 1999 год. Фирма Intel выпустила процессор Pentium III, по-строенный на базе Pentium II с поддержкой инструкции SSE (Streaming SIMD Extensions).
* 2000 год. Компания Microsoft выпустила операционную системуWindows 2000.
* 2000 год. Фирмы Intel и AMD объявили о выпуске процес-соров с тактовой частотой 1 ГГц.
* 2000 год. Фирма Intel анонсирует процессор Itanium - пер-вый процессор семейства P7.

369



**Приложение 2. История развития МП Intel**

"Intel - это процессоры"- Энди Гроув (Andy Grove). До на-чала 70-х годов вычислительные машины были доступны весьма ограниченному кругу специалистов, а их применение, как правило, оставалось окутанным завесой секретности и мало известным ши-рокой публике. Однако в 1971 г. произошло событие, которое в корне изменило ситуацию и с фантастической скоростью превра-тило 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, полу-чивший название "МИКРОПРОЦЕССОР".

Сегодня, оглядываясь на 25 лет назад, мы не очень погре-шили бы против истины, добавив к этому слову титул " Его Вели-чество", - столь велико оказалось влияние крошечного полупро-водникового кристалла практически на все сферы деятельно-сти.Именно ему мы обязаны появлением нового класса вычисли-тельных 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-

370

тупны - слишком дорого тогда стоило машинное время мэйн-фреймов и мини-ЭВМ) лишь системам, занимавшим не одну сот-ню квадратных метров. Наверное, никогда прежде человек не имел в своих руках инструмента, обладающего столь колоссальной мо-щью 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, призван-ных заменить ЗУ на магнитных сердечниках. Поскольку к концу 60-х годов память этого типа практически исчерпала весь свой по-тенциал развития, проблема была весьма актуальной, а ее разра-ботка сулила немалые прибыли. И хотя в данной области Intel до-билась заметных успехов, тем не менее мировую славу ей принес-ли совсем другие изделия. Поворотным моментом в истории ком-пании стал 1969 г., когда был получен заказ на создание ряда спе-циализированных микросхем для калькуляторов от ныне уже не-существующей японской фирмы Busicom. В апреле того же года в штаб- квартиру Intel прибыли три инженера из Busicom, среди ко-торых 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 пре-допределившего дальнейший путь развития корпорации Intel. Вы-полнение заказа японской компании поручили Марциану Хоффу (Marcian Hoff), в то время ведущему специалисту Intel. Проанали-зировав техническое задание (ТЗ), а также предложенный Шима вариант архитектуры и системы команд , Хофф пришел к выводу, что поставленную задачу можно решить и более простыми средст-вами, оптимизировав систему команд и дополнив схему устройст-ва модулем памяти. Окончательная версия ТЗ была им разработана совместно с другим инженером компании, Стэном Мэзором (Stan Mazor), и предусматривала функционирование процессора на так-товой частоте 1 МГц. Любопытно, что на такой же частоте работал и компьютер модели 1620 фирмы IBM, но при этом ориентиро-вочная стоимость изделия Intel составляла 30 - 40 дол., а одна

371

только арендная плата за пользование IBM 1620 доходила до 2000дол. в месяц. Темп работы над прибором был настолько вы-сок, что уже осенью 1969 г. заказчику представили проект кри-сталла. Поскольку вариант, предложенный Intel, оказался более универсальным и имел потенциал для применения не только в калькуляторах, Busicom отдала предпочтение американской разра-ботке. Для японского менеджмента того времени подобное реше-ние 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 Фэгги-ну (Federico Faggin), который и осуществил это менее чем за год. Кстати, спустя всего лишь несколько лет он станет одним из осно-вателей и президентом фирмы Zilog, которая получит широкую известность благодаря микропроцессору Z80, и поныне выпускае-мому рядом изготовителей. Вместе с тем, несмотря на успешный ход работ, одно обстоятельство беспокоило многих проектиров-щиков 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 всту-пить с Intel в переговоры о снижении цен на поставляемые микро-схемы (в связи с обострением конкуренции на рынке калькулято-ров), Хофф рекомендовал сотрудникам отдела маркетинга согла-ситься 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 проекти-ровщиков от программирования логических функций в их обору-довании. А на вопрос: "Будете ли вы делать это, если цена соста-вит 5 дол.?" - был получен столь же краткий, сколь и категорич-ный ответ: "Безусловно". Резюме маркетинговому исследованию

372

подвел Боб Нойс: "Все, что нам надо - это довести стоимость комплекта интегральных схем до 30 или 40 дол., и тогда покупате-ли 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 при-менении.

Это натолкнуло команду Гелбаха на идею создания систем проектирования микроЭВМ , содержащих по крайней мере про-стейшие инструментальные средства разработки и отладки про-граммного обеспечения. Идея оказалась настолько плодотворной, что уже через несколько лет поставки подобных систем стали при-носить доход , соизмеримый с доходом от продажи самих микро-процессоров. В результате кристалл вышел на рынок в сопровож-дении соответствующих средств поддержки и началось триум-фальное 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 микропро-цессора Intel 4004 - именно такое обозначение получил первый прибор, послуживший отправной точкой абсолютно новому классу полупроводниковых устройств.

Кристалл представлял собой 4-разрядный процессор с клас-сической архитектурой ЭВМ гарвардского типа и изготавливался по передовой в те годы p-канальной МОП- технологии с проект-ными 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 де-шифратором 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 однокристального процессора i4004 имел весьма впечатляю-щие характеристики. Память команд достигала 4Кбайт (для срав-нения: объем ЗУ мини-ЭВМ в начале 70-х годов редко превышал

373

16 Кбайт), а регистровый файл ЦП насчитывал шестнадцать 4 - разрядных регистров, которые можно было использовать и как во-семь 8-разрядных (восемь 4-разрядных пар). Такая организация РОНов сохранена и в последующих микропроцессорах фирмы Intel. Три регистра стека обеспечивали три уровня вложения под-программ. Конечно, эта цифра не вызвала особых восторгов у про-граммистов, тем не менее они получили возможность создавать полноценные программы. Процессор i4004 монтировался в пласт-массовый 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. рассчита-на на широкий круг решаемых задач и разрабатываемых приложе-ний. Первоначальное назначение кристалла наложило определен-ный отпечаток на состав системы команд, поэтому присутствие в ней ряда инструкций, в частности десятичной коррекции, а также наличие соответствующих аппаратных средств не вызывает особо-го удивления.

Вместе с тем кристалл располагал весьма ограниченными средствами ввода/ вывода, а в системе команд отсутствовали опе-рации 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 при-бор, 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 выво-дами - самый распространенный (а значит, и самый дешевый) тип корпуса в начале 70- х годов. А поскольку в распоряжении инже-неров оказался узкий интерфейс с "внешним миром", то пришлось пойти на применение мультиплексированной шины адреса и дан-ных, причем 12- разрядный адрес выдавать порциями по четыре

374

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 повест-ку дня этот больной вопрос. Задача создания многовыводных кор-пусов оказалась крепким орешком и попортила немало крови кон-структорам и технологам, не говоря уж о самих разработчиках процессоров, которых к тому времени число "40", наверное, при-водило просто в бешенство.

Примерно к середине 70-х годов появились корпуса типов

DIP и QUIP (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. Пласт-массовые 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 вы-водов и приемлемой ценой была решена в начале 80-х годов. Од-нако все это было впереди, а в самом начале 70-х пионерам при-шлось исходить из существовавших реалий и принимать соответ-ствующие технические решения. Говоря о первом в мире микро-процессоре, нельзя не вспомнить и о том, что с самого начала спе-циалисты позаботились о простоте и удобстве построения систем на базе i4004. Компанией был разработан и выпущен не один кри-сталл 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

375

на совместное использование , разработка аппаратных средств системы заметно упрощалась, и это стало не последней причиной популярности i4004. i4004 + обработка прерываний + ... = i4040.

Опыт использования первого МП показал, что такие факто-ры, как отсутствие средств обработки прерываний, наличие трех уровней вложения подпрограмм и необходимость реализации ло-гических операций И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ с помощью специальных подпрограмм , далеко не всегда удовлетворяют раз-работчиков. Создав новый рынок и захватив на нем господствую-щие высоты, Intel тем не менее стремилась расширить его грани-цы, и решение о выпуске усовершенствованного варианта i4004, свободного от указанных недостатков, было вполне логичным хо-дом компании. Использование же в новом кристалле всех нарабо-ток, 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 возмож-ности предшествующей модели и существенно улучшены как тех-нические, так и программные средства. Система команд пополни-лась 14 инструкциями , включая выполнение логических операций И и ИЛИ; кроме того, в процессор были введены средства остано-ва 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 спе-циальных команд. Отчасти такая организация обусловлена тем, что процессор теперь мог обращаться к двум блокам памяти ко-манд объемом 4 Кбайт и за каждым из них программист мог за-крепить свою область регистров. Наряду с этим восемь РОНов бы-ли всегда доступны для использования. В итоге получилась доста-точно гибкая и удобная структура, позволявшая разрабатывать са-мостоятельные программные модули , способные взаимодейство-вать через общую часть регистрового файла. Обработка одноуров-невых прерываний - одно из наиболее существенных новшеств чипа i4040 - превратила его в полноценный процессор и сделала возможным использование в системах реального масштаба време-ни. Благодаря применению сигнала "останов" стала реальностью

376

синхронизация работы процессора с некоторыми внешними со-бытиями.

Вместе с тем специальных операций со стеком (запись в стек содержимого РОН и извлечение его из стека) разработчики ПО так и не дождались. Несмотря на то что тактовая частота и машинный цикл i4040 не претерпели изменений , производитель-ность процессора возросла за счет использования более совершен-ной архитектуры и эффективной системы команд. 60 инструкций, ориентированных на широкий спектр решаемых задач, обработка прерываний, до 8 Кбайт памяти команд, а также возможность бы-строго перевода систем на базе i4004 на новый процессор вывели i4040 в безусловные лидеры рынка 4-разрядных устройств. Прав-да, полной совместимости с программным обеспечением, разрабо-танным 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 от-вергли новое детище Intel. Преимущества, которые i4040 имел пе-ред своим предшественником, с лихвой компенсировали отмечен-ный 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 продук-ции, которое они начнут испытывать спустя всего 5 - 7 лет, а по-тому могли себе позволить такую роскошь, как частичная несо-вместимость 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 по-коления 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 ап-реля 1972 г. начала поставки первого в отрасли 8-разрядного при-бора 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-

377

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). Проектирова-ние i8008 шло практически параллельно с работами над i4004. Ко-гда 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 ре-шила пустить i8008 в свободную продажу и в очередной раз не ошиблась. Кристалл изготавливался по p-канальной МОП-технологии с проектными нормами 10 мкм и содержал 3500 тран-зисторов. Процессор работал на частоте 500кГц при длительности машинного цикла 20 мкс (10 периодов задающего генератора).

В отличие от своих предшественников новый МП имел ар-хитектуру ЭВМ принстонского типа, а в качестве памяти допускал применение комбинации ПЗУ и ОЗУ . Помимо увеличения разряд-ности и перехода на использование общего поля памяти для ко-манд и данных, структура процессора претерпела еще ряд сущест-венных изменений. Прежде всего это коснулось регистрового фай-ла 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 па-мяти. В связи с этим следовало бы ожидать снижения производи-тельности, которого на самом деле не произошло, поскольку опе-рации 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 и воз-можностями технологии , которая в то время еще не позволяла размещать на кристалле большие регистровые структуры (анало-гично кристаллам i4004 и i4040 в МП i8008 блок РОНов был реа-лизован в виде динамической памяти, необходимость регенерации

378

которой влечет за собой применение ряда дополнительных ап-паратурных средств).

Почти вдвое (с восьми до пяти состояний) сократилась дли-тельность 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 медлен-ными устройствами был введен сигнал готовности (READY). Раз-работчики 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 похва-статься не могли.

Система команд первого 8-разрядного ЦП насчитывала 65 инструкций, причем значительно увеличилось число команд ус-ловных переходов, а также логических инструкций и команд сдви-га. Новый кристалл мог адресовать память объемом до 16 Кбайт (объем ЗУ для ЦП типа i4040 не превышал 8 Кбайт). Его произво-дительность по сравнению с 4-разрядными системами возросла в 2,3 раза. Процессор с такими параметрами уже можно было рас-сматривать как серьезную заявку на многие очень перспективные секторы рынка, включая контрольно- испытательное оборудова-ние, прецизионную измерительную технику и сложные промыш-ленные контроллеры систем управления технологическими про-цессами.

Однако i8008 получил от своих предшественников и "тяже-лое 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 ми-ром" ограничил количество управляющих сигналов процессора: в результате специалистам Intel пришлось использовать их шифра-цию, что повлекло за собой необходимость установки дополни-тельного внешнего оборудования для формирования сигналов управления. В среднем для сопряжения процессора с памятью и устройствами ввода/вывода требовалось около 20 схем средней степени интеграции.

379

Если спортсмены исповедуют древний олимпийский принцип "Citius, altius, fortius", то разработчикам он ближе в не-сколько измененном виде: "Citius, citius, citius". Вскоре после вы-хода 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 опера-тивной памяти , послужило лучшей рекомендацией для активного использования кристалла в различных областях, начиная от про-мышленности и медицины и кончая военной электроникой и тор-говлей. По мере расширения сферы влияния микропроцессора и усложнения систем на его базе возросли и требования к нему со стороны проектировщиков оборудования. Программное обеспече-ние уже с трудом вписывалось в 16 Кбайт, да и производитель-ность прибора начинала "поджимать" многих разработчиков. Кро-ме того, некоторые области применения настойчиво требовали расширения не только количества , но и номенклатуры периферий-ных 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 распро-странение. Гордон Мур, один из основателей и первый президент Intel, в те годы писал, что микропроцессоры найдут еще более ши-рокое применение, если улучшить их рабочие характеристики, ибо уже выявились области, где требуются предельно высокие пара-метры.

Возможности p-канальной МОП -технологии для создания сложных высокопроизводительных МП были уже практически ис-черпаны, поэтому направление главного удара перенесли на тех-нологию n-МОП. Перед проектировщиками стояли не менее слож-ные проблемы - разработка эффективной системы команд, рассчи-танной на широкий круг решаемых задач, при сохранении про-граммной совместимости с предыдущей моделью, расширение объема адресуемой памяти, поддержка интенсивного ввода/ вывода без существенной потери производительности процессора, совер-

380

шенствование подсистемы обработки прерываний. i8080 - три-умф 8-разрядных систем Работа над новым 8-разрядным процес-сором 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 ветера-нами "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 повыше-нию 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 проектирова-ния, разработчики попытались использовать маски i8008 в произ-водстве чипов ОЗУ 2102. Однако, как вспоминает Стэн Мэзор, по-сле 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 архитек-туры, 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 ком-пьютер, который стоил в то время тысячи долларов. Поэтому мы чувствовали, что предложили вполне разумную цену". По свиде-тельству Эда Гелбаха, расходы на исследования и разработки по программе i8080 окупились в течение первых пяти месяцев про-даж. Новый ЦП практически по всем статьям разительно отличал-ся от своих предшественников. Благодаря использованию техно-логии n-МОП с проектными нормами 6 мкм, на кристалле удалось разместить 6 тыс. транзисторов. При этом геометрические размеры

381

самого кристалла по сравнению с i8008 увеличились незначи-тельно. Следовательно, процент выхода годных изделий и ряд экономических показателей производства, включая себестоимость, удалось сохранить на достаточно высоком уровне. Тактовая часто-та 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 пред-шествующей модели. К великой радости системщиков и програм-мистов объем памяти, адресуемой процессором, был увеличен в четыре раза и достиг 64 Кбайт (кстати, в то время ОЗУ такой ем-кости предлагали потребитетелям минимальные конфигурации многих мини-ЭВМ). В сочетании с эффективным механизмом об-работки прерываний это давало им карт-бланш для широкого при-менения нового МП в сложных системах сбора и обработки ин-формации 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 до-полнительных внешних схем для разделения потоков адресов и данных. Общее же количество микросхем, требовавшихся для по-строения 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 программ-но-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 объ-ем ПО, сохранение программной совместимости i8080 и i8008 бы-ло вполне естественным и разумным шагом компании (таким об-разом, Intel встала на тот же путь, что и корпорация IBM с компь-ютерами знаменитой серии System 360, оказавшись на долгие годы своего рода заложницей собственного творения). Именно поэтому в состав РОНов нового процессора были включены основные ра-бочие регистры предыдущей модели. Правда, полной совместимо-сти с i8008 достичь опять не удалось, так как процедуры обраще-

382

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 про-цессора на новый в некоторых случаях программы приходилось полностью перерабатывать. Включение в систему команд ряда ин-струкций, адресующих память с использованием трех пар регист-ров (в i8008 для этого выделялась одна пара), придало дополни-тельную гибкость системе и существенно упростило жизнь про-граммистам, реализация же блока РОНов на основе статической, а не динамической памяти дала дополнительную экономию площа-ди кристалла для размещения других схем процессора. Исключе-ние 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 па-мятью и АЛУ во время выполнения арифметических и логических операций.

Новым веянием в архитектуре микропроцессоров стало ис-пользование многоуровневой системы прерываний по вектору. Та-кое техническое решение позволило довести общее число источ-ников прерываний в системе до 256. Правда, до появления специа-лизированных БИС контроллеров прерываний схема формирова-ния векторов прерываний требовала применения до десяти допол-нительных чипов средней степени интеграции. В отличие от пре-рываний 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 па-мять емкостью до 64 Кбайт, сэкономив тем самым ОЗУ для раз-мещения программ и данных. Тот факт, что разработчики микро-процессоров 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 ис-пользовано и кто станет первым. Пионером в этом опять оказалась

Intel.

Освобождение центрального процессора от управления внешними устройствами и обмен данными между памятью систе-

383

мы и периферией, минуя ЦП, были уже достаточно давно и ус-пешно 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 микро-процессоров по большим системам. ПДП открыл зеленую улицу для применения в микроЭВМ таких сложных устройств, как нако-пители на магнитных дисках и лентах, а также дисплеи на ЭЛТ, которые и превратили микроЭВМ в полноценную вычислитель-ную 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 пополнили БИС контроллера ПДП и контроллера пре-рываний. Благодаря хорошо продуманному составу комплекта, проектирование микро-ЭВМ на его базе в ряде случаев упрости-лось настолько, что было подобно сборке домика из детских куби-ков. Мимо такой техники пройти было трудно! Мощная система команд, высокое быстродействие, простота проектирования, про-тотипные комплекты и системы разработки вскоре превратили микропроцессор в стандарт де-факто. Более того, конкуренты от-стали 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 полупро-водниковых 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 распростране-нию 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. Полу-проводниковые компании , выпускавшие микропроцессоры собст-венной разработки, росли как грибы после дождя, и не было прак-

384

тически ни одной мало- мальски уважающей себя фирмы, кото - рая не попробовала бы свои силы на этой стезе . Безусловно, все они отбирали у Intel определенную долю рынка, но самыми серь-езными 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 пре-зидент 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 обра-щения 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, ко-торые существенно упростили построение сложных систем, а так-же ЦП i8085, уже свободного от указанных недостатков , имевшего ряд преимуществ перед конкурентами и обладавшего программ-ной совместимостью со своим предшественником. Но, пожалуй, самым сильным ответным ходом компании стало создание сис-темного 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 пред-решило исход борьбы, и последствия той победы компания, по-видимому, ощущает до сих пор. Вторая половина 70-х годов была отмечена массированной атакой японских поставщиков полупро-водниковых устройств на рынок США. Объемы производства аме-риканских компаний росли, а прибыли падали. Пожалуй, наи-большему давлению подвергся сектор ЗУ, которые занимали не

385

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 мик-ропроцессорных комплектов.

Операционная система iRMX-80 стала для этих машин ба-зовым системным ПО. Таким образом, из изготовителя микросхем корпорация превратилась в производителя подсистем, обеспечив себе тем самым дополнительный стабильный рынок сбыта полу-проводниковых изделий . Именно с тех пор в лексикон "кремние-вых" компаний прочно вошел термин OEM (Original Equipment Manufacturer - изготовитель комплексного оборудования).

Ориентацию на перспективу можно смело отнести к харак-терной особенности стиля Intel Corporation. Вероятно, нельзя на-звать ни одного прибора, работа над которым не проводилась бы одновременно с проектированием изделия следующего поколения (достаточно вспомнить историю создания кристалла i8008). Сле-дуя установившейся традиции, в конце 1974 г. фирма приступила к созданию прототипа 32-разрядной системы iAPX-432, на которую возлагались большие надежды. Рост производительности планиро-валось получить не только благодаря увеличению разрядности, но и за счет использования сложной архитектуры с возможностью ор-ганизации мультипроцессорной системы.

Следует отметить, что мультипроцессорные универсальные ЭВМ в ту пору еще не стали привычным атрибутом вычислитель-ных центров и многие нюансы построения таких машин были из-вестны 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 объек-тивной 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-

386

дство компании сочло целесообразным попытаться развить ус-пех i8080, и в самом начале 1976 г. стартовали работы по созда-нию 16-разрядного прибора, который впоследствии получил обо-значение i8086.

Конечной целью нового проекта было получение 16-разрядного микропроцессора с производительностью, на порядок превышающей аналогичный параметр кристалла i8080. Постав-ленная задача решалась за счет дальнейшего совершенствования архитектурных концепций, положенных в основу его предшест-венника. Избранная стратегия эволюционного, а не революцион-ного 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 техниче-ского директора компании по микропроцессорам. Коллектив , ядро которого составили Боб Кохлер (Bob Koehler), Джон Бэйлис (John Bayliss), Джим Маккевит (Jim McKevitt), Чак Уайлдмэн (Chuck Wildman) и Стив Морз (Steve Morse), возглавил Билл Полмэн (Bill Pohlman).

Время и давление конкурентов были критическими факто-рами, 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 на-чала выполнения проекта!). Прибор изготавливался по высокока-чественной трехмикронной МОП-технологии с кремниевыми за-творами (H-MOS), позволившей разместить на чипе 29 тыс. тран-зисторов. Высокое быстродействие элементов ( задержка 2 нс/вентиль) обеспечило тактовую частоту процессора 5 МГц, а 16-разрядная архитектура и 200-нс машинный цикл - производитель-ность процессора, превыщающую аналогичный параметр i8080 на порядок величины.

Программная совместимость с i8080/i8085 была, пожалуй, единственной, но вместе с тем и исключительно важной характе-ристикой, которая объединяла 86-й кристалл с его предшественни-ками. Структура процессора оказалась полностью пересмотрен-

387

ной. Прежде всего, прибор был разбит на два функциональных блока (рис. 3) - операционный (Execution Unit, EU) и интерфейс-ный (Bus Interface Unit, BIU), которые могли работать одновре-менно. В результате исполнение одной команды совмещалось во времени с выборкой следующей инструкции или данных из памя-ти. Более того, в ЦП появился небольшой буфер команд, что дава-ло дополнительную экономию времени при обращениях к памяти. Так, микропроцессоры позаимствовали у универсальных ЭВМ еще одно техническое решение, и именно с этого момента в них нача-лась реализация принципов параллелизма.

Возможность адресации 1- Мбайт ОЗУ и сегментация памя-ти могут быть отнесены к одним из наиболее существенных нов-шеств, предложенных инженерами Intel. В частности , сегментация памяти и большое число уровней прерываний были ориентирова-ны на работу систем в многозадачном режиме , весьма актуальном для приложений управления. (Правда, следует отметить, что меха-низм защиты памяти реализован не был, и в ряде случаев это су-щественно усложняло разработку ПО). Большая емкость ОЗУ по-зволяла перевести проекты построения сложных операционных и прикладных систем из области теории в сферу практической реа-лизации.

Наряду с поддержкой ввода/вывода по каналу прямого дос-тупа к памяти чип i8086 обеспечивал адресацию до 64К портов программно-управляемого ввода/вывода. Это снимало практиче-ски любые ограничения при формировании крупных систем сбора и обработки информации. Исключительно удачной находкой мож-но считать два режима работы процессора - минимальный и мак-симальный. 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 модифи-кации своих продуктов.

Однако, пожалуй, наиболее интересна система команд процессора. 147 инструкций позволяли решать задачи управления практически

388

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 кри-сталл 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 сопровожде-нии средств поддержки: вспомогательных БИС, средств разработ-ки и отладки аппаратуры и системного ПО, а также прототипных комплектов и одноплатных микроЭВМ серии iSBC86.

Несмотря на блестящие характеристики продукта и мощ-ную поддержку со стороны изготовителя, потребовалось почти два года, чтобы кристалл завоевал признание разработчиков. Первое время даже его создатели стали опасаться - не слишком ли слож-ное детище они сотворили. Однако феномену медленного роста объема продаж нашлось разумное толкование: начальные темпы и не могли быть высокими, поскольку создаваемой системе требует-ся новое системное и прикладное ПО. Справедливость такого объ-яснения подтвердилась в начале 1980 г., когда закончился лабора-торный 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. Ком-пания Motorola извлекла уроки из битвы за сектор 8-разрядных систем и спустя примерно год после появления чипа i8086 пред-ложила 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 по-коления МП. Поэтому усилия были сосредоточены на ускорении выпуска процессора с 8-разрядной внешней шиной данных (i8088) и реализации плана агрессивного маркетинга приборов, получив-шего название Operation Crush. В основу кампании, которую воз-

389

главил Билл Дэвидоу (Bill Davidow), были положены два мо-мента: во-первых, поставка пользователю готовых решений, вклю-чая системное ПО (ОС iRMX-86), широкий набор периферийных БИС и оперативную техническую поддержку; во- вторых, подго-товка потребителей к будущим техническим решениям на базе платформы Intel. В соответствии с новой стратегией претерпел из-менения и традиционный рекламный девиз компании, вполне за-кономерно 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 практиче-ского использования микропроцессорного комплекта i8086. Про-светительская деятельность сопровождалась массированной рек-ламной кампанией. Акция имела грандиозный успех - в течение года процессор i8086 стал базовым элементом более 2 тыс. проек-тов. 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 от-крытую систему и базироваться на стандартном для отрасли мик-ропроцессоре.

Работы шли в г. Бока-Ратон (шт. Флорида) . К этому момен-ту на рынке присутствовали кристаллы i8086, i8088, Z8000 и M68000. Сотрудники Intel с замиранием сердца ждали решения "Голубого гиганта". Дэйв Хаус, тогда генеральный менеджер ком-пании по микропроцессорам и периферийным кристаллам, вспо-минает, 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 ника-ких официальных заявлений на этот счет. Поскольку Apple выбра-ла для своих машин кристалл M68000, успокаиваться было рано и кампания Operation Crush была продолжена. И только после полу-чения от IBM первых крупных заказов на процессоры стало окон-чательно 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

390

i8086 в девять раз превысили соответствующий показатель для

M68000.

Использование микросхем i8086 в IBM PC предопределило дальнейшее развитие корпорации Intel как разработчика и изгото-вителя универсальных процессоров общего назначения. Вычисли-тельная мощь 16-разрядных приборов была поддержана арифме-тическим сопроцессором i8087, который позволил превратить ПК в достаточно мощный инструмент и для решения задач вычисли-тельного характера . Более того, теперь и разработчики систем управления на базе 86-го ЦП получили возможность использовать интенсивную арифметическую обработку информации, для кото-рой ранее служили мини -ЭВМ. Бурный рост объемов продаж пер-сональных компьютеров явно свидетельствовал о том, что рожда-ется новый рынок огромной емкости.

В конце 70 - начале 80-х годов основными сферами приме-нения микропроцессоров продолжали оставаться различные сис-темы управления (ПК еще не появились). Сложность решаемых задач требовала высокой производительности ЦП, большого объ-ема ОЗУ и поддержки многозадачных сред, характерных для по-добного рода комплексов. Для разработчиков программного обес-печения 1 Мбайт оперативной памяти достаточно быстро превра-тился в своеобразное прокрустово ложе, а отсутствие в i8086 аппа-ратных средств поддержки защиты памяти существенно усложня-ло создание ПО. Решение этих задач и было основной целью раз-работки следующего кристалла.

Процессор i80286 был анонсирован 1февраля 1982 г. и сра-зу приковал к себе пристальное внимание специалистов. Архитек-тура и характеристики чипа оказались весьма впечатляющими. Оставшись 16- разрядным прибором, по производительности но-вый ЦП в 3 - 6 раз превзошел своего предшественника при такто-вой частоте первой модификации 8 МГц. Благодаря использова-нию многовыводного корпуса (к тому времени проблема их мас-сового производства была успешно решена) разработчики смогли применить схему с раздельными шинами адресов и данных. 24 разряда адреса позволили обращаться к физической памяти объе-мом до 16 Мбайт - такую же емкость имели и старшие модели большинства мэйнфреймов. Встроенная система управления памя-

391

тью и средства ее защиты открывали широкие возможности ис-пользования 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 защищен-ный. В первом случае он воспринимался как быстрый ЦП i8086 с несколько расширенной системой команд и прекрасно подходил тем потребителям, для которых, помимо скоростных характери-стик , жизненно важным было сохранение существующего задела ПО. Работа в защищенном режиме позволяла использовать пре-имущества прибора в полном объеме, и прежде всего - большой объем основной памяти. Этими качествами решила воспользовать-ся IBM, применив процессор в новой модели ПК типа AT, что с неимоверной 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 разработчи-кам разместить на кристалле контроллеры прерываний и ПДП, а также таймер и системный генератор. Таким образом, покупателю были предложены мощные однокристальные ЭВМ, с помощью ко-торых можно было реализовать законченную систему с числом ИС не более десяти. Уже в первый год после начала поставок количе-ство проданных чипов i80186 превысило в 30 раз аналогичный по-казатель для i8086.

Спрос на i80286 также требовал постоянного увеличения объемов его производства. Длина слова - 32 разряда Использова-ние 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.

392

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 про-цессор - 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 ре-шить. (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 состоя-лась, но сам факт появления подобного рода информации свиде-тельствует о достаточно тесных контактах обеих компаний. Не ис-ключено, что корпорация IBM выступала в качестве заказчика кристалла следующего поколения, а ее инженеры принимали не-посредственное 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 проект-ными нормами 1 мкм и двумя уровнями металлизации позволило разместить на кристалле 275 тыс. транзисторов и реализовать пол-ностью 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 сис-тема управления памятью и защиты включала регистры преобра-зования адреса, механизмы защиты оперативной памяти и улуч-шенные аппаратные средства поддержки многозадачных ОС. По-мимо работы с виртуальной памятью допускались операции с па-мятью, имевшей страничную организацию. Предварительная вы-борка команд, буфер на 16 инструкций, конвейер команд и аппа-ратная 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 аналогич-ный показатель для МП i8086. Безусловно, новый прибор остался

393

совместимым со своими предшественниками на уровне объект-ных кодов. Одной из наиболее любопытных особенностей рас-сматриваемой разработки компании было использование кэш-памяти , позволившей существенно повысить производительность систем на базе 386-го процессора (еще один атрибут универсаль-ных машин, который стал применяться в микропроцессорных сис-темах).

Для управления работой буфера Intel создала БИС высоко-производительного контроллера кэш-памяти типа i82385, с помо-щью которой формировался двухвходовой множественный ассо-циативный кэш. Указанная БИС обеспечивала управление буфе-ром емкостью до 32 Кбайт и высокий коэффициент удачных об-ращений. Поскольку степень интеграции была еще недостаточна для реализации на том же кристалле и математического сопроцес-сора, он по-прежнему выпускался в виде отдельного кристалла i80387, дополняя вычислительную мощь ЦП.

Особый интерес представляли три режима работы кристал-ла - 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 расширенными системой ко-манд и регистрами. Благодаря этим качествам 32- разрядного про-дукта компания сохранила прежних клиентов, которые хотели мо-дернизировать свои системы, не отказываясь от имевшегося задела в области программного обеспечения, и привлекла тех, кому изна-чально требовалась высокая скорость обработки информации. Этот "двунаправленный" маркетинговый ход затронул как разра-ботчиков и пользователей ПК, так и специалистов, занимающихся сложными системами управления. Одно из основных ограничений реального режима было связано с предельным объемом адресуе-мой памяти, равным 1 Мбайт. От него свободен защищенный ре-жим, позволяющий воспользоваться всеми преимуществами архи-тектуры нового ЦП. Размер адресного пространства в этом случае увеличивался до 4 Гбайт, а объем поддерживаемых программ - до

394

64 Тбайт. Производителям ПО это позволяло задействовать достаточно гибкие методы разработки и создавать более крупные программные пакеты.

Для конечных пользователей выполнение приложений, рас-считанных на работу в реальном и защищенном режимах, проис-ходило без каких- либо функциональных отличий, поскольку управление обоими режимами базировалось на средствах ОС и специальном прикладном ПО. Однако системы защищенного ре-жима обладали более высоким быстродействием и возможностями организации истинной многозадачности. Наконец, режим вирту-ального 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 не-сколько 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 аппа-ратуры, можно было вместо нескольких ЦП типа i8086 использо-вать 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 форми-ровании технических требований к первому 32-разрядному МП не только IBM (о чем уже говорилось), но и Microsoft. И хотя впо-следствии пути этих компаний разошлись (достаточно упомянуть противостояние OS/2 и Windows), процессор одинаково успешно работал с первыми версиями обеих систем. Стремление удовле-творить запросы потребителей всех категорий привело руково-дство Intel к идее повторить свой удачный опыт с процессором i8088, выпустив вариант 386- го МП с 16- разрядной внешней ши-ной данных (при сохранении внутренней 32-разрядной архитекту-ры). Существующий прибор получил обозначение i80386SX и был анонсирован 16 июня 1988 г., а уже менее чем через полгода поль-зователям были предложены первые ПК на его основе. Поскольку

395

эти модели стоили дешевле компьютеров с ЦП 80386DX, мно - гие потребители вполне справедливо рассматривали их как на-чальную ступень в применении вычислительной техники.

Так Intel создала еще один сектор рынка. В конце 80-х годов сте-пень интеграции микросхем быстро приближалась к 1млн транзи-сторов 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 Гор-дону Муру вдохновения, чтобы придумать ей достойное примене-ние.

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 позво-лили разработчикам реализовать на кристалле не только кэш-память , но и математический сопроцессор. Такое техническое ре-шение свело к возможному минимуму число чипов на плате и са-мым благоприятным образом сказалось на стоимости готовых сис-тем. В отличие от имевшегося в МП 80386 двухвходового множе-ственного ассоциативного кэша в i80486 использовался более эф-фективный четырехвходовой буфер, который, будучи размещен-ным в чипе, мог работать на тактовой частоте процессора. Как и предыдущий МП, новый прибор функционировал в трех режимах

* был ориентирован на многозадачные среды. За счет интеграции математического сопроцессора в БИС, а также модернизации его архитектуры производительность на задачах вычислительного ха-рактера 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,

* через два года модели ПК на его базе полностью вытеснили из-делия предыдущего поколения. В течение шести лет кристалл без-раздельно господствовал на рынке. Институт "вторых поставщи-ков", возникший еще в эпоху 8-разрядных систем, продолжал су-

396

ществовать, и изготовлением микросхем класса 486 занимались такие гиганты электронной промышленности, как IBM, Texas Instruments и AMD. На совершенствовании архитектуры кристал-лов Intel родились и сделали имя Сyrix и NexGen. Конкуренты ока-зались достаточно шустрыми и за ними требовался присмотр. По-этому, даже приступив к проработкам процессора следующего по-коления, компания не оставляла свое последнее творение без вни-мания.

Ровно через два года после выпуска i80486 появилась уп-рощенная версия кристалла ( без сопроцессора), получившая по да-леко не очевидной аналогии с предшествующей моделью обозна-чение i80486SX. Дальнейшее совершенствование пошло по пути увеличения тактовой частоты : были представлены версии на 50, 66, 75 и 100 МГц. Разработку вариантов с более высокими такто-выми частотами сочли нецелесообразной, и компании-конкуренты получили в этом отношении карт-бланш. Intel же была целиком устремлена в будущее.

Pentium - пятое поколение микропроцессоров Стремитель-ное усложнение программного обеспечения и постоянное расши-рение сферы применения компьютеров настоятельно требовали существенного роста вычислительной мощи центральных процес-соров ПК. Ко всему прочему на пятки стали наступать и RISC-процессоры. Хотя в конце 80-х годов некоторые эксперты пред-сказывали близкий конец кристаллов CISC, корпорация Intel впол-не справедливо посчитала, что до этого еще далеко и в микропро-цессорах использованы не все возможности CISC-архитектуры. Кроме того, фирме вряд ли простили бы отказ от программной со-вместимости с предшествующими моделями - стоимость накоп-ленного системного и прикладного ПО уже измерялась в миллиар-дах долларов.

Как это случалось не раз, проработки нового процессора начались , когда проект создания 486-го МП вступил в заключи-тельную стадию ( начало 1989 г.). Менеджером команды "архитек-торов" кристалла следующего поколения был назначен Дональд Альперт (Donald Alpert) из отделения микропроцессоров корпора-ции Intel, а работы по проектированию арифметического устрой-ства с плавающей точкой возглавил Дрор Эвнон (Dror Avnon).

397

В основу продукта была положена суперскалярная архитектура (еще один атрибут из мира мэйнфреймов), которая и дала возмож-ность получить пятикратное повышение производительности по сравнению с моделью 486DX.

Высокая скорость выполнения команд достигалась благо-даря двум 5-ступенчатым конвейерам, позволявшим одновременно исполнять несколько инструкций. Для постоянной загрузки обоих конвейеров из кэша требуется широкая полоса пропускания. Со-вмещенный буфер команд и данных обеспечить ее не мог, и разра-ботчики воспользовались решением из арсенала RISC-процессоров, оснастив Pentium раздельными буферами команд и данных. При этом обмен информацией с памятью через кэш дан-ных осуществлялся совершенно независимо от процессорного яд-ра, а буфер инструкций был связан с ним через высокоскоростную 256-разрядную внутреннюю шину. Несмотря на то что новый кри-сталл был спроектирован как 32-разрядный , для связи с остальны-ми 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 графи-ческой информации, мультимедиа-приложения, а также все более интенсивное использование ПК для решения задач вычислитель-ного характера требуют высокой производительности при выпол-нении операций с плавающей точкой. Скоростные характеристики этого блока получены за счет применения конвейерной архитекту-ры устройства (8-ступенчатый конвейер FPU позволил выдавать результаты каждый такт), а также благодаря аппаратной реализа-ции (вместо традиционной микропрограммной) основных арифме-тических операций. Сложение, умножение и деление выполнялись раздельными высокопроизводительными блоками умножителя, сложителя и делителя. Средства для работы с плавающей точкой поддерживали стандарт IEEE 754 и операции с одинарной, двой-ной и расширенной (80-битовый формат) точностью, а также на-бор базовых трансцендентных функций. В итоге по скорости вы-

398

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 парамет-рами прибора, подогрела к нему интерес как поставщиков компь-ютеров, так и пользователей. Пристальное внимание Intel тради-ционно уделила тесному взаимодействию с разработчиками ПО, которые получили спецификации процессора Pentium одними из первых и очередные версии приложений проектировали уже с уче-том возможностей кристалла. Таким образом, процессор вышел в свет в сопровождении почетного эскорта новых программных продуктов.

Несмотря на ряд проблем, практически всегда возникаю-щих 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 не-скрываемое раздражение некоторых крупных потребителей про-дукции 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, необходимо-сти в утверждении архитектуры, и без того давно ставшей миро-вым 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 ста-новилось 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

399

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 про-межуточные барьеры в 66; 75; 90; 100; 120; 133 и 166 МГц, в этом году Intel довела тактовую частоту Pentium II - представитель вто-рого 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 двухотсековом керамическом корпусе вместе со специализированным кэшем L2, а представляет собой традиционную однокристальную конструкцию, в которой предусмотрены отдельные соединения с системной шиной и ин-терфейс кэша L2. Однако Intel не продает этот кристалл сам по се-бе; процессор устанавливается на небольшой печатной плате вме-сте 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. Го-раздо важнее то, что благодаря двукратному уменьшению скоро-сти 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, выпус-каемых другими изготовителями, высвобождает производствен-ные мощности фирмы Intel, а благодаря острой конкуренции на рынке SRAM сохраняется низкий уровень себестоимости. Это по-зволило Intel сделать 512-Кбайт кэш L2 стандартной принадлеж-ностью Pentium II; вариант 512-Кбайт кэша для Pentium Pro обхо-дится дорого и используется очень редко. В результате хотя, ин-

400

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 двух-процессорные 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 од-но- и двухпроцессорных серверах будет применяться Pentium Pro, поскольку для выполнения прикладных задач, требующих обра-ботки больших объемов данных, преимущества высокоскоростно-го кэша L2 могут оказаться гораздо важнее, нежели большая ем-кость кэшей и высокая тактовая частота. Кроме того, серверы, как правило, мало что выигрывают от технологии ММХ. 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 (ин-

тегрированная)

Тип корпуса: Organic Lan Grid Array 603 (OLGA 603)

Частота системной шины: 400 МГц Потоковые SIMD расширения SSE2

Область применения: Двухпроцессорные серверы и рабочие стан-ции

**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 (ин-

тегрированная)

401

Тип корпуса: PGA423, PGA478

Частота системной шины: 400 МГц Потоковые SIMD расширения SSE2

Область применения: офисные и домашние ПК, одно- и двухпро-цессорные 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 (интег-рированная)

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 KБ Advanced Transfer Cache (ин-

тегрированная) 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 двухпро-цессорные 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

402

Разрядность системной шины: 64 разрядов Адресуемая память: 64 ГБ

Область применения: четырех-восьмипроцессорные рабочие стан-ции

**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 (ин-

тегрированная)

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 двухпроцес-сорные серверы и рабочие станции

**Процессор Intel Pentium III c технологией сверхнизкого энер-гопотребления**

Дата выпуска: 13 ноября 2001 г. Тактовая частота: 700 МГц Частота системной шины: 100 МГц

403

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 производительно-сти и 1.15 в режиме оптимальной производительности Напряжение питания: 2 В в режиме оптимальной производитель-ности 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 производительно-сти и 1.05 в режиме оптимальной производительности Напряжение питания: 1 В в режиме оптимальной производитель-ности Область применения: полный спектр мобильных ПК

**Процессор Pentium III cо сверхнизким энергопотреблением и технологией Intel SpeedStep≥ (600, 500 МГц) для мобильных**

404

**ПК**

Дата выпуска: 21 мая 2001 г. (600 МГц); 30 января 2001 г. (500 МГц)

Тактовая частота: 600, 500, 300(режим оптимальной производи-тельности) 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 оптималь-ной 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 c низким энергопотреблением и техно-**

**логией Intel SpeedStep≥ (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 производитель-ности Область применения: офисные и пользовательские мобильные ПК

**Процессоры для мобильных ПК**

**Процессор Pentium III c технологией Intel SpeedStep≥ (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-

405

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 c технологией Intel SpeedStep≥ (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 (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

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 (ин-

тегрированная)

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

406

**Для встраеваемых применений**

**Процессор Pentium III (700 МГц) для встраеваемых примене-ний**

Дата выпуска: 19 марта 2001 г.

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (ин-

тегрированная)

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 техноло-гии 0.18-микрон**

Дата выпуска: 24 мая 2000 г. (933 МГц)

Кэш-память второго уровня: 256 КБ Advanced Transfer Cache (ин-

тегрированная)

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 двухпро-цессорные сервера

**Процессор Pentium III (1 ГГц, 933, 866/850 МГц), произведен-**

**ный 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 (ин-

тегрированная)

407

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 двухпро-цессорные 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 (ин-

тегрированная)

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 двухпро-цессорные 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 двухпро-цессорные сервера

**Процессор Pentium II Xeon(450 МГц)**

408

Объявлен: 5 января 1999 г. Тактовая частота: 450 МГц

Данные по производительности: **серверы**, **рабочие станции**

Кэш L2: 512 KБ, 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 KБ

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 коруса: Картридж с односторонним контактом (S.E.C) Частота шины: 100 МГц

Ширина полосы пропускания шины: 8 bytes

409

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 MГц)**

Объявление: 14 июня 1999 г.

Тактовая частота: 400 MГц; **данные по производительности**

Количество транзисторов: 27.4 млн (0.18-мкм процесс), 256 KБ 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 MГц)**

Объявление: 14 июня 1999 г.

Тактовая частота: 400 MГц; **данные по производительности**

Количество транзисторов: 27.4 млн (0.25-мкм процесс), 256 KБ 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 MГц)**

Объявление: 25 января 1999 г.

Тактовые частоты: 266, 300, 333 и 366 MГц; **данные по произво-**

**дительности**

Количество транзисторов: 27.4 млн (0.25-мкм процесс), 256 KБ встроенной кэш L2

Тип корпуса: Ball Grid Array (BGA): число выводов = 615

Габариты: ширина = 31 мм; длина = 35 мм Напряжение ядра: 1.6 В

410

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 K 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 K 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 яд-

ро 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)

411

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 двухпроцес-сорные серверы и рабочие станции

**Процессор Pentium II (350 и 400 MГц)**

Объявлен: 15 апреля 1998 г. Тактовая частота: 350, 400 МГц;

Количество транзисторов: 7.5 млн (0.25-мкм процесс), кэш L2 512K

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 MГц

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 кор-

рекцией 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

412

**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 Kб

Ширина полосы пропускания шины: 64-битная системная шина с ECC; 64-битная шина кэш с оптимизацией ECC

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 бизне-са, рабочие станции и серверы

**Процессор Intel Celeron**

**Технология сверхнизкого энергопотребления**

**Мобильный процессор Celeron со сверхнизким энергопотреб-лением и технологией Intel SpeedStep≥ (600 и 500 МГц)**

Объявлены: 21 мая 2001 г. (600 МГц); 30 января 2001 г.

Тактовые частоты: 500 и 600 МГц; **данные по производительно-**

**сти**

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≥ (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)

413

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 второ-го 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 производительно-**

**сти**

Количество транзисторов: 18,9 млн (0,25-мкм технология), кэш-память второго уровня 128 Kб

Тип корпуса: Ball Grid Array (BGA), 615 выводов Габариты: ширина = 31 мм; длина = 35 мм Напряжение ядра: 1.9 В

Питание: 466 MГц = 15.6 Вт; 433 MГц = 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**

414

Количество транзисторов: 18,9 млн (0,25-мкм технология), кэш-память второго уровня 128 Kб

Тип корпуса: Ball Grid Array (BGA), 615 выводов Габариты: ширина = 31 мм; длина = 35 мм Напряжение ядра: 1.6 В

Питание: 400 MГц = 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 Kб

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 Kб

Тип корпуса: Ball Grid Array (BGA), 615 выводов Габариты: ширина = 31 мм; длина = 35 мм Напряжение ядра: 1.6 В

Питание: 333 MГц = 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 производительно-**

**сти**

Количество транзисторов: 18.9 млн (0.25-мкм процесс), 128K встроенной кэш L2

Тип корпуса: Ball Grid Array (BGA), число выводов = 615

415

Габариты: ширина = 31 мм; длина = 35 мм Напряжение ядра: 1.6 В

Питание: 300 MГц = 7.7 Вт; 266 MГц = 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 уров-ня: 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 производитель-**

**ности**

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

416

**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 выходов

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

417

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 (300A 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 MГц)**

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: 66MГц

Разрядность шины: 64 -bit системная шина Адресуемая память: 4 Гигабайт Виртуальная память: 64 Терабайт

418

Габариты модуля: 5" x 2.275" x .208"

Применение: базовые ПК

**Процессор Intel Celeron (266 MГц)**

Объявлен: 15 апреля 1998 г.

Тактовая частота: 266 MГц; **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 встро-енной 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 Kб кэш-памяти второго уровня) Количество транзисторов: 5,5 млн (0,35-мкм технология), 256 КБ кэш-памяти второго уровня 256 Kб: 15,5 млн (0,6-мкм технология),

419

512 Kб кэш-памяти второго уровня: 31 млн (0,35-мкм техноло-гия)

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≥**

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: Tape Carrier Package (TCP), 320 выводов Применение: мобильные ПК и ноутбуки

**Процессор Pentium (266 МГц) с технологией MMX≥**

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: Tape Carrier Package (TCP), 320 выводов Применение: мобильные ПК и ноутбуки

**Процессор Pentium (200, 233 МГц) с технологией MMX≥**

Объявлены: 8 сентября 1997 г. Тактовая частота: 200, 233 МГц

Количество транзисторов: 4,5 млн (0,25-мкм технология)

420

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≥**

Объявлен: 2 июня 1997 г.

Тактовая частота: 233 МГц (7.12 SPECint95, 5.21 SPECfp95. Пока-затель тестирования iCOMP≥ Index 2.0 - 203).

Количество транзисторов: 4,5 млн (0,35-мкм технология КМОП) Ширина полосы пропускания шины: 64 бит (внешняя шина дан-ных), 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 вы-водами - PPGA)

Габариты модуля: 5 см x 5 см

Применение: высокопроизводительные настольные ПК и серверы

**Процессор Pentium (200, 166 МГц) с технологией MMX≥**

Объявлен: 8 января 1997 г.

Тактовая частота: 200, 166 МГц (6.44 SPECint95, 4.87 SPECfp95.

Показатель тестирования iCOMP≥ Index 2.0 - 182).

Количество транзисторов: 4,5 млн (0,35-мкм технология КМОП) Ширина полосы пропускания шины: 64 бит (внешняя шина дан-ных), 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 вы-водами - PPGA)

Габариты модуля: 5 см х 5 см Применение: высокопроизводительные настольные ПК и серверы

421

**Процессор Pentium (200 МГц)**

Объявлен: 10 июня 1996 г.

Тактовая частота: 200 МГц (5.17 SPECint95,4.32 SPECfp95. Пока-затель тестирования iCOMP≥ Index 2.0 - 142).

Количество транзисторов: 3.3 млн (0,35-мкм технология БиКМОП) Ширина полосы пропускания шины: 64 бит (внешняя шина дан-ных), 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 вы-водами - PPGA)

Габариты модуля: 5 см х 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 Xxpress с 1 Mб кэш-памяти второго уровня)

Количество транзисторов: 3,3 млн (0,35-мкм технология БиКМОП) Ширина полосы пропускания шины: 64 бит (внешняя шина дан-ных), 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 х 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 Xxpress с 1 Mб кэш-памяти второго уровня)

Количество транзисторов: 3,3 млн (0,35-мкм технология БиКМОП) Ширина полосы пропускания шины: 64 бит (внешняя шина дан-ных), 32 бит (адресная шина)

422

(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 х 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 Xxpress с 1 Mб кэш-памяти второго уровня)

Количество транзисторов: 3,2 млн (0,6 и 0,35-мкм технологии БиК-МОП)

Ширина полосы пропускания шины: 64 бит (внешняя шина дан-ных), 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 х 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 Kб кэш-

памяти второго уровня)

100 МГц (166,3 млн. операций в секунду, 3.30 SPECint95, 2.59 SPECfp95 на системе Xxpress с 1 Mб кэш-памяти второго уровня) Количество транзисторов: 3,2 млн (0,6-мкм технология БиКМОП) Ширина полосы пропускания шины: 64 бит (внешняя шина дан-ных), 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)

423

Габариты модуля: 5 см х 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 Kб кэш-

памяти второго уровня)

Количество транзисторов: 3,2 млн (0,6-мкм технология БиКМОП) Ширина полосы пропускания шины: 64 бит (внешняя шина дан-ных), 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, располо-женных в шахматном порядке (SPGA)

Габариты модуля: 5 см х 5 см

TCP: 2,4 х 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 Kб кэш-памяти второго уровня)

66 МГц (112 млн. операций в секунду , 77.9 SPECint92, 63.6 SPECfp92 на системе Xpress с 256 Kб кэш-памяти второго уровня) Количество транзисторов: 3,1 млн (0,8-мкм технология БиКМОП) Ширина полосы пропускания шины: 64 бит (внешняя шина дан-ных), 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 х 5,49 см Применение: настольные ПК

424

**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 Mбайт

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 Kб кэш-

памяти второго уровня)

Процессор с тактовой частотой 66 МГц, о начале выпуска которо-го объявлено 10 августа 1992 г. (54 млн. операций в секунду, 39.6 SPECint92, 18.8 SPECfp92 на системе Micronics M4P с 256 Kб 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.

425

Тактовая частота : 16 МГц, начало выпуска процессора объяв-лено 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 тех-нология)

Ширина полосы пропускания шины: 32 бит Адресуемая память: 4 гигабайт Виртуальная память: 64 терабайт

Применение: недорогие настольные ПК начального уровня класса

Intel486≥.

Аналог процессора Intel486≥ DX, но без встроенного математиче-ского сопроцессора.

Имеется возможность модернизации путем установки процессора

Intel OverDrive.

Этот процессор стал типовым при работе со встроенными прило-жениями.

**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 предназначен-ный для портативных ПК Высокоинтегрированная архитектура включает в себя кэш-память, шину и средства управления памятью

426

**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. опера-

ций в секунду, 22.4 SPECint92 на системе Micronics M4P с 128 Kб

кэш-памяти второго уровня)

50 МГц - начало выпуска объявлено 24 июня 1991 г. (41 млн. опе-

раций в секунду, 33.4 SPECint92, 14.5 SPECfp92 на системе

Compaq/50L с 256 Kб кэш-памяти второго уровня)

Количество транзисторов: 1 200 000 (1 мкм , в процессорах с такто-вой 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 процес-сора 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

427

**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. опе-раций 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 Kб 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 доста-точно 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 процес-сора занал 45 секунд

428

**80186**

Объявлен: 1982 г.

Примечание: применялся, главным образом, в работе с управляю-щими 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 производ-ства 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 вычисле-ние цены товара.

429

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 Kбайт

Применение: устройства управления уличным освещением, ком-

пьютеры 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 процес-сором 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 Kбайт

Применение: терминалы воода-вывода, калькуляторы общего на-значения, 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

430

**Приложение 4. Первая десятка самых мощных микропроцессорных систем (суперкомпьютеров)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №№ | Произво- | Система | Год | Число |
|  | дитель |  |  | 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 пло-щадки и установленный в Ливерморской национальной лаборато-рии. Он включает 512 SMP-узлов на базе 64-разрядных процессо-ров POWER3-II (в общей сложности 8192 процессора) и использу-ет новую коммуникационную технологию Colony с пропускной

431

способностью около 500 Мбайт/с, что почти в четыре раза бы-

стрее коммутатора SP high-performance switch.

Архитектура суперкомпьютера основана на зарекомендо-вавшей себя массивно-параллельной архитектуре RS/6000 и обес-печивает производительность в 12,3 Тфлопс (триллионов операций в секунду). Система включает в общей сложности 8 Тбайт опера-тивной 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. Су-перкомпьютер используется учеными Министерства энергетики США для расчета сложных трехмерных моделей с целью поддер-жания ядерного оружия в безопасном состоянии. ASCI White со-стоит из трех отдельных систем, среди которых самой большой является White (512 узлов, 8192 процессора), а есть еще Ice (28 уз-лов, 448 процессоров) и Frost (68 узлов, 1088 процессоров).

Предшественником ASCI White был суперкомпьютер Blue Pacific (другое название ASCI Blue), включающий 1464 четырех-процессорных узла на базе кристаллов PowerPC 604e/332 МГц. Узлы связаны в единую систему с помощью кабелей общей дли-ной почти в пять миль, а площадь машинного зала составляет 8 тыс. квадратных футов. Система ASCI Blue состоит в общей слож-ности из 5856 процессоров и обеспечивает пиковую производи-тельность в 3,88 Тфлопс. Суммарный объем оперативной памяти составляет 2,6 Тбайт.

Американский национальный центр по исследованию атмо-сферы (NCAR) выбрал IBM в качестве поставщика самого мощно-го в мире суперкомпьютера, предназначенного для прогнозирова-ния климатических изменений. Система, известная под именем Blue Sky (Синее небо) на порядок увеличивает возможности NCAR в области моделирования климата. Ядром Blue Sky являют-ся суперкомпьютер IBM SP и системы IBM eServer p690, примене-ние которых позволяет добиться пиковой производительности почти в 7 Тфлопс при объеме дисковой подсистемы IBM SSA в 31,5 Тбайт.

432



Кабельное соединение узлов суперкомпьютера ASCI White

Суперкомпьютер, получивший название "Синий шторм" (Blue Storm), создается по заказу Европейского центра средне-

срочных 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 процессо-ров. В 2004 г. "Синий шторм" оснащен серверами нового поколе-ния p960, которые делают его еще в два раза мощнее. Суперком-пьютер работает под управлением ОС AIX. Общая емкость нако-пителей Blue Storm составляет 1,5 петабайт, вычислительная мощь - около 23 Тфлопс. Система весит 130 т, а по мощи в 1700 раз пре-восходит шахматный суперкомпьютер Deep Blue.

Исследователи IBM совместно с Ливерморской националь-ной лабораторией ведут работы над компьютерами Blue Gene/L и Blue Gene/C. Эти компьютеры - часть начатого еще в 1999 г. с це-лью изучения белков 5-летнего проекта Blue Gene, в который было вложено 100 млн долл. Проектная производительность Blue Gene/L будет, таким образом, превышать суммарную производи-тельность 500 самых мощных компьютеров в мире. При этом но-вый суперкомпьютер занимает площадь, равную всего половине теннисного корта. Инженеры IBM поработали и над снижением потребления энергии - его удалось уменьшить в 15 раз.