Тема №6 Функционирование ЭВМ

Содержание и порядок проведения лекции	Время, мин
Вступительная часть	5
Основная часть (текст лекции)	
Введение	3
Учебные вопросы:	
1. Общая характеристика 32-разрядных процессоров.	30
2. Структурная схема ядра и принципы функционирования 32-разрядного процессора.	45
Заключение	5
Заключительная часть	2

Текст лекции

Введение

Объявить тему, цели, порядок проведения занятия.

Учебные вопросы

1. ОБЩАЯ ХАРАКТЕРИСТИКА 32-РАЗРЯДНЫХ ПРОЦЕССОРОВ

Процессоры более поздних, чем Pentium, разработок (от Pentium Pro до Pentium 4 включительно) обладают некоторыми общими свойствами:

1). Вторичная статическая КЭШ-память (размером **512 Кб**, ... , **2Мб**,...) соединена с процессором специально выделенной шиной (32-разрядной у Pentium II, Pentium III; у Pentium 4 – всегда 64-разрядной).

Кристалл КЭШ2 содержит в несколько раз больше транзисторов, чем кристалл центрального процессора.

Столь большое число транзисторов в КЭШе объясняется его статической природой. Статическая память обычно использует шесть транзисторов для запоминания одного бита, в то время как в динамической памяти было бы достаточно одного транзистора на бит.

Статическая память быстрее, но дороже.

Производительность вычислительной системы, построенной на быстром процессоре, очень сильно зависит от точного соответствия процессору микросхем его окружения, в частности вторичного КЭШа. Далеко не все фирмы-производители компьютеров могут данное соответствие обеспечить.

То есть вторичный КЭШ должен быь настроен на процессор оптимальным образом, что облегчает проектирование материнской платы.

Первые процессоры с выделенной шиной обращались к вторичному КЭШу по данной шине с той же (собственной) тактовой частотой (Pentium Pro). Однако с ростом тактовой частоты процессора стало слишком сложно и дорого поддержи-

вать такую частоту на материнской плате. Поэтому стали применяться делители частоты.

Использование специально выделенной шины для доступа к вторичному КЭШу улучшает производительность вычислительной системы.

а) достигается полная синхронизация скоростей процессора и шины;

б) исключается конкуренция с другими операциями ввода-вывода и связанные с этим задержки.

Шина КЭШа второго уровня полностью отделена от внешней шины, через которую происходит доступ к памяти и внешним устройствам. 64-битовая внешняя шина (например, PCI) может работать со скоростью, равной половине, одной третьей или одной четвертой от скорости процессора, при этом шина вторичного КЭШа работает независимо от этого на полной скорости.

Подобная архитектура называется **архитектурой двойной независимой шины** или **DIB**-архитектурой (**D**ual **I**ndepended **B**us). Её введение сразу же удвоило скорость обмена данными.

2). Тенденция к увеличению числа стадий конвейера.

Процессор **Pentium** со своей конвейерной и суперскалярной архитектурой содержит два 5-стадийных конвейера, которые могут работать параллельно и выполнять две целочисленные команды за машинный такт. При этом параллельно может выполняться только пара команд, следующих в программе друг за другом и удовлетворяющих определенным правилам.

В **Pentium Pro**, **Pentium II**, **Pentium III** для увеличения производительности осуществлен переход к одному **12**-стадийному конвейеру. Увеличение числа стадий приводит к уменьшению выполняемой на каждой стадии "работы" и, как следствие, к уменьшению времени нахождения команды на каждой стадии на 33 процента по сравнению с **Pentium**.

В процессоре **Pentium 4** число стадий конвейера доведено уже до **20** (максимальное число стадий на сегодняшний день).

Причина появления спроектированного с чистого листа процессора Pentium 4 - это ничто иное, как ограничения, накладываемые на тактовую частоту процессоров семейства P6 - из-за конвейера длиной 12 ступеней, причем ступеней относительно больших, которые при технологическом процессе, которым располагает Intel, не в состоянии работать на частотах, существенно превышающих 1 Гигагерц. Предел данного технологического процесса находится где-то в районе 1,1 - 1,2 ГГц.

В основе **Pentium 4** лежит архитектура, названная Intel NetBurst architecture. Этим названием Intel хотел подчеркнуть, что основная цель нового процессора – ускорить выполнение задач потоковой обработки данных, напрямую связанных с бурно развивающимся Internet.

3). Динамическое исполнение команд.

Возможности суперскалярной архитектуры **Pentium** с ее способностью к выполнению двух команд за такт было бы трудно превзойти без совершенно нового подхода.

Тщательный анализ факторов, ограничивающих производительность современных микропроцессоров, доказал, что при выполнении реальных программ мощность процессоров не используется в полной мере.

Например, при выполнении команды загрузки из памяти в регистр может оказаться, что содержимое соответствующей ячейки памяти отсутствует в КЭШе. При традиционном подходе процессор перейдет к выполнению следующей команды только после того, как данные из ячейки основной памяти будут прочитаны. Все время ожидания процессор будет простаивать.

В то время как скорость процессоров за последние 10 лет выросла по меньшей мере в 10 раз, время доступа к основной памяти уменьшилось только на 60 процентов. Это увеличивающееся отставание скорости работы с памятью по отношению к скорости процессора и явилось той фундаментальной проблемой, которую необходимо было решить.

Один из возможных подходов к решению этой проблемы - разработка высокопроизводительных компонентов, окружающих процессор. Однако массовый выпуск систем, включающих и высокопроизводительный процессор, и высокоскоростные специализированные микросхемы окружения, был бы слишком дорогостоящим.

Второй вариант - увеличение размера КЭШа второго уровня, чтобы уменьшить процент случаев отсутствия необходимых данных в КЭШе.

Это решение эффективное, но тоже чрезвычайно дорогостоящее, особенно учитывая требования по времени доступа к компонентам КЭШа второго уровня.

Решение сформулированной выше проблемы заключается в так называемом **динамическом исполнении команд**, которое может быть охарактеризовано как оптимальное (квазиоптимальное) выполнение программы, основанное на предсказании будущих переходов, анализе графа потоков данных с целью выбора наилучшего порядка исполнения команд и на опережающем выполнении команд в выбранном оптимальном порядке.

Примененный впервые в **Pentium Pro** подход устраняет жесткую зависимость между традиционными фазами "выборки" и "выполнения", когда последовательность прохождения команд через эти две фазы соответствует последовательности команд в программе.

Новый подход связан с использованием так называемого **пула команд** и с новыми эффективными методами предвидения будущего поведения программы.

При этом традиционная фаза "выполнение" заменяется на две: "диспетчеризация/выполнение" и "откат".

В результате команды могут начинать выполняться в произвольном порядке (так называемое опережающее выполнение), но завершают свое выполнение всегда в соответствии с их исходным порядком в программе.

Процессор **Pentium II**, если говорить кратко, представляет собой модификацию **Pentium Pro** с поддержкой **MMX**, но без интегрированной КЭШ-памяти второго уровня. Конструктивно **Pentium II** размещается на дочерней плате (там же располагается вторичная КЭШ-память), которая устанавливается в специальный разъем на системной плате.

В **Pentium II** также увеличена до **32К** КЭШ-память первого уровня (по **16К** на КЭШ команд и КЭШ данных), а вторичный КЭШ, в отличие от **Pro**, работает на половинной частоте процессора.

Кроме того, были добавлены схемы ускорения выполнения 16-битных приложений.

Процессор **Pentium III** изготовлен уже по технологии **0,25 мкм** и содержит более **9,5 миллионов** транзисторов на кристалле. Он унаследовал все качества своих предшественников: динамическое исполнение команд, архитектуру **DIB**, технологию **MMX**, и приобрел ряд новых.

В него были включены **70** новых **SIMD**-инструкций, ускоряющих работу с приложениями трехмерной графики, потокового аудио, видео и распознавания речи, а кроме того, у **Pentium III** появилась функция серийного номера, который, являясь уникальным идентификатором процессора, может быть использован для определения компьютера в сети или со стороны прикладных программ.

2. СТРУКТУРНАЯ СХЕМА ЯДРА И ПРИНЦИПЫ ФУНКЦИОНИРОВАНИЯ 32-РАЗРЯДНОГО ПРОЦЕССОРА

2.1 Структурная схема ядра процессора.

На рисунке 1 приведена структурная схема ядра процессора.

Прежде чем приступить к ее рассмотрению, введем понятия "упорядоченного" и "беспорядочного" устройства.

Будем понимать под "упорядоченным" - устройство, которое работает в соответствии с исходным порядком команд в программе, а под "беспорядочным" - устройство, которое не учитывает исходный порядок команд в программе.

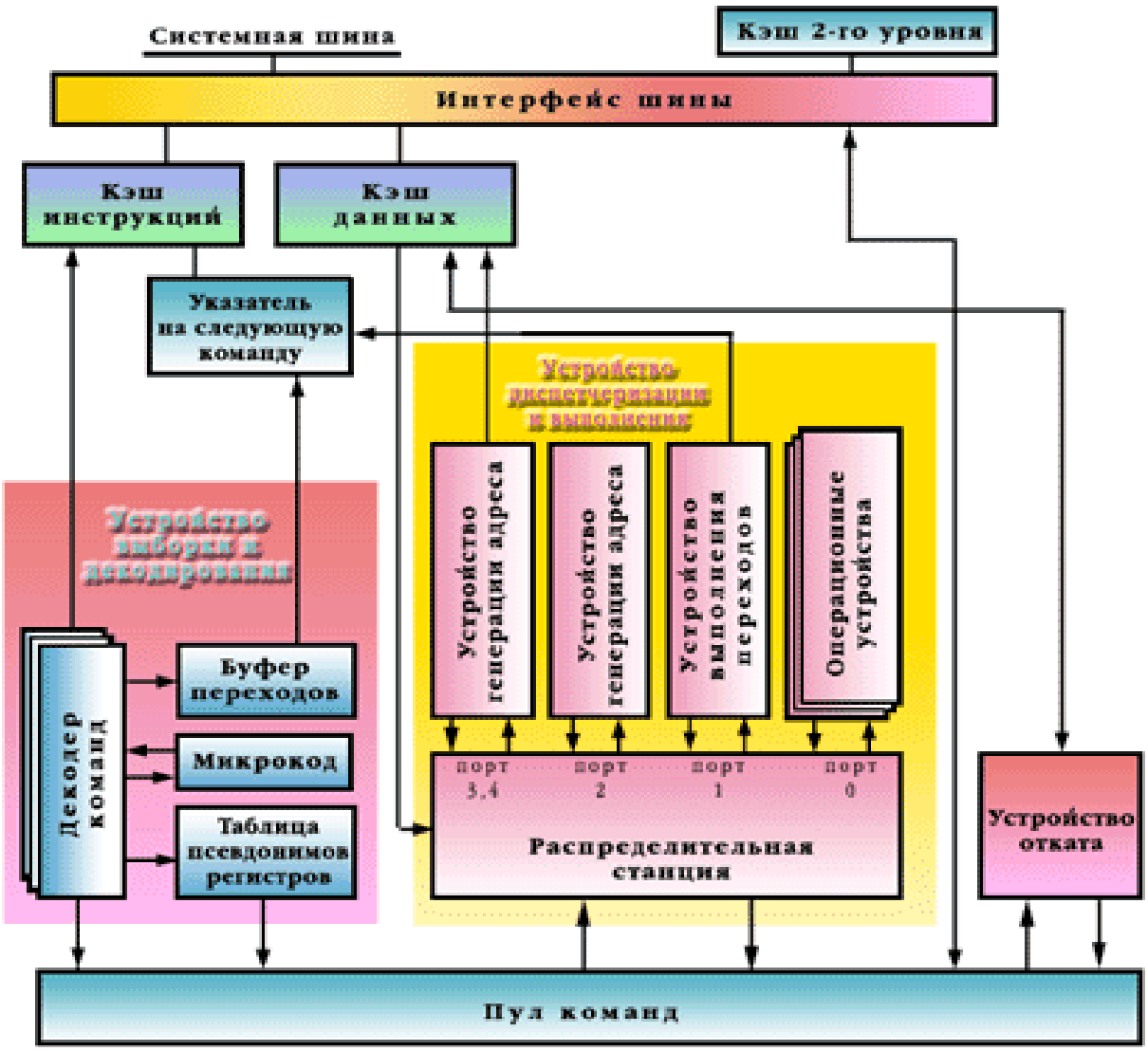

Рис.1. Структурная схема ядра процессора

1). Устройство выборки/декодирования является "упорядоченным" устройством, которое воспринимает на входе поток команд из программы пользователя и декодирует их, превращая в последовательность микрокоманд, соответствующих потоку данных в программе пользователя.

Три параллельных декодера команд предназначены для приема потока выбранных из КЭШа байтов команд, их обработки и декодирования. Декодер преобразует команды архитектуры Intel в микрокоманды.

Большинство команд преобразуются в одну микрокоманду, сложные же команды требуют обращения к микрокоду, представляющему из себя набор заранее составленных последовательностей микрокоманд.

Таблица псевдонимов регистров служит для преобразования ссылок на логические регистры в ссылки на физические регистры.

Буфер переходов с 512 входами использует расширенный алгоритм **Йе** (**Yeh**), который обеспечивает, по утверждению Intel, более чем 90-процентную точность предсказания переходов.

Указатель на следующую команду - это индексная память КЭШа команд, содержимое которой определяется буфером переходов, состоянием процессора и

сообщениями о неправильном предсказании перехода, поступающими из устройства диспетчеризации/выполнения.

Индекс в указателе на следующую команду адресует строку КЭШа команд и следующую за ней строку.

2). Устройство диспетчеризации/выполнения является "беспорядочным" устройством, которое воспринимает поток данных и планирует выполнение микрокоманд с учетом зависимостей по данным и доступности ресурсов, а также временно сохраняет результаты опережающего выполнения в пуле команд.

Пятипортовая распределительная станция обеспечивает взаимодействие с вычислительными ресурсами устройства диспетчеризации/выполнения.

В случае доступности операндов и необходимого для выполнения микрокоманды устройства выполнения операций (вычислительный ресурс), распределительная станция выбирает микрокоманду из пула команд и направляет ее на устройство для выполнения. Результаты выполнения микрокоманды возвращаются в пул.

Устройство выполнения переходов предназначено для обеспечения исключения из пула команд всех микрокоманд, загруженных после команды перехода, в случае ошибочного его предсказания. Оно также обеспечивает формирование правильного адреса перехода и его передачу в буфер переходов для перезапуска конвейера с нового адреса.

Устройства генерации адресов предназначены для формирования адресов обращений к соответствующим регистрам и основной памяти.

3). Устройство отката - "упорядоченное" устройство, которое определяет момент и порядок завершения выполнения команды, то есть перевод временных результатов опережающего выполнения в постоянное состояние вычислительной системы.

Оно на основании проверки статуса отыскивает в пуле команд микрокоманды, которые уже выполнены, с целью их последующего удаления из пула.

Именно при удалении микрокоманды результаты ее выполнения, хранящиеся в пуле команд, реально изменяют состояние вычислительной системы.

Удаление микрокоманд из пула осуществляется таким образом, чтобы изменение состояния вычислительной системы соответствовало первоначальному порядку команд в программе.

Помимо этого, устройство отката учитывает и обрабатывает прерывания и другие исключительные ситуации.

Устройство отката также контролирует реальный набор регистров процессора.

Дело в том, что небольшое их количество в архитектуре процессоров "Intel" приводит к интенсивному использованию каждого из них и, как следствие, к возникновению множества зависимостей между командами, использующими один и тот же регистр.

Поэтому, чтобы исключить задержку в выполнении команд из-за данных зависимостей, устройство диспетчеризации/выполнения работает с дублями реги-

стров, находящимися в пуле команд (одному регистру может соответствовать несколько дублей).

4). Интерфейс шины является "частично упорядоченным" устройством, отвечающим за связь трех вышеупомянутых устройств с внешним миром.

Интерфейс шины взаимодействует непосредственно с КЭШем 2-го уровня и поддерживает до 4 параллельных обращений к нему.

Интерфейс шины также управляет обменом данными с основной памятью с использованием протокола MESI.

5). Пул команд реализован в виде массива контекстно-адресуемой памяти, называемого также буфером переупорядочивания. Он служит для хранения всех подлежащих выполнению микрокоманд вместе с дополнительной информацией о их статусе, а также временного запоминания результатов опережающего их выполнения.

2.2. Принципы функционирования процессоров.

Из КЭШа команд выбирается строка, соответствующая индексу в указателе на следующую команду, и следующая за ней строка, после чего выбранные 16 байтов поступают на декодеры.

Чаще всего длина строки КЭШ1 – 32 байта. Предвыборка инструкций –16-байтными блоками, выровненными по 16-байтным границам.

Две строки считываются из-за того, что команды в архитектуре Intel выровнены по границе байта, и поэтому может происходить передача управления на середину или конец строки КЭШа.

Выполнение этой ступени конвейера занимает три такта, включая время, необходимое для вращения предвыбранных байтов и их подачи на декодеры команд.

Начало и конец команд помечаются.

Три параллельных декодера принимают поток отмеченных байтов и обрабатывают их, отыскивая и декодируя содержащиеся в потоке команды.

Декодеры преобразуют команды архитектуры Intel в микрокоманды, которые ставятся в очередь и затем посылаются в таблицу псевдонимов регистров, где ссылки на логические регистры преобразуются в ссылки на физические регистры, после чего каждая из микрокоманд вместе с дополнительной информацией о ее состоянии (статусе) посылается в пул команд.

На этом заканчивается "упорядоченная" часть конвейера.

Устройство диспетчеризации/выполнения выбирает микрокоманды из пула команд в зависимости от их **статуса**. Под статусом понимается информация о доступности операндов микрокоманды и наличии необходимых для ее выполнения вычислительных ресурсов.

Если статус микрокоманды показывает, что ее операнды уже вычислены и доступны, а необходимое для ее выполнения вычислительное устройство также

доступно, она выбирается из пула команд и направляется на устройство выполнения операций. Результаты выполнения микрокоманды возвращаются в пул.

Взаимодействие с вычислительными ресурсами происходит через пятипортовую распределительную станцию.

Процессор может запускать на выполнение до 5 микрокоманд за такт, по одной на каждый порт. Средняя длительно поддерживаемая пропускная способность - 3 микрокоманды за такт.

Процесс планирования выполнения микрокоманд является принципиально "беспорядочным": момент направления микрокоманд на вычислительные ресурсы определяется только потоками данных и доступностью ресурсов, без какой бы то ни было связи с первоначальным порядком команд в программе.

Если в каждом такте для каждого устройства готова к выполнению только одна микрокоманда, то проблемы выбора не возникает.

Но если готовых к выполнению на данном устройстве микрокоманд несколько, то используется алгоритм планирования FIFO ("первый пришел - первый обслужен").

Алгоритм, реализованный в буфере переходов, позволяет в большинстве случаев правильно предсказать, состоится или не состоится переход, но иногда все же предсказания будут ошибочны.

Для исправления случаев неверного предсказания перехода применен следующий подход:

Микрокомандам перехода еще в упорядоченной части конвейера ставятся в соответствие адрес следующей команды и предполагаемый адрес перехода. После вычисления перехода реальная ситуация сравнивается с предсказанной.

Если они совпадают, то проделанная исходя из предположения об исходе перехода работа оказывается полезной, так как соответствует реальному ходу программы, а микрокоманда перехода удаляется из пула команд.

Если же допущена ошибка (переход был предсказан, но не произошел, или было предсказано отсутствие перехода, а в действительности он состоялся), то устройство выполнения переходов изменяет статус всех микрокоманд, засланных в пул команд после команды перехода, чтобы убрать их из пула команд.

Правильный адрес перехода направляется в буфер переходов, который перезапускает весь конвейер с нового адреса.

Устройство отката проверяет статус микрокоманд в пуле команд, определяет те микрокоманды, которые уже выполнены и могут быть удалены из пула. При удалении микрокоманды результаты ее выполнения, хранящиеся в пуле команд, реально изменяют состояние вычислительной системы, например, происходит запись в регистры.

Изменение состояния вычислительной системы строго соответствует первоначальному порядку команд в программе.

Процесс отката занимает два такта. В первом такте устройство отката считывает пул команд и отыскивает готовые к откату микрокоманды; затем оно определяет, какие из этих микрокоманд могут быть удалены из пула в соответствии с исходным порядком команд в программе.

Во втором такте результаты отката записываются в пул команд и в регистровый файл отката. Устройство отката может обработать три микрокоманды за такт.

Таким образом, реализованная комбинация таких архитектурных методов, как улучшенное предсказание переходов (почти всегда правильно определяется предстоящая последовательность команд), анализ потоков данных (определяется оптимальный (квазиоптимальный) порядок выполнения команд) и опережающее выполнение (предвиденная последовательность команд выполняется без простоев в оптимальном порядке), позволила увеличить производительность при использовании той же самой технологии производства.

Эта комбинация методов и получила название "динамическое выполнение".