

Архитектура ЭВМ и систем

Лекция № 7

Современное состояние и
тенденции развития процессоров

План лекции

1. Архитектурные особенности процессоров Pentium
2. Программная модель процессоров Pentium
3. Аппаратная организация защиты в процессорах Pentium
4. Аппаратные средства поддержки многозадачности
5. Перспективы развития процессоров

Современное состояние и тенденции развития процессоров

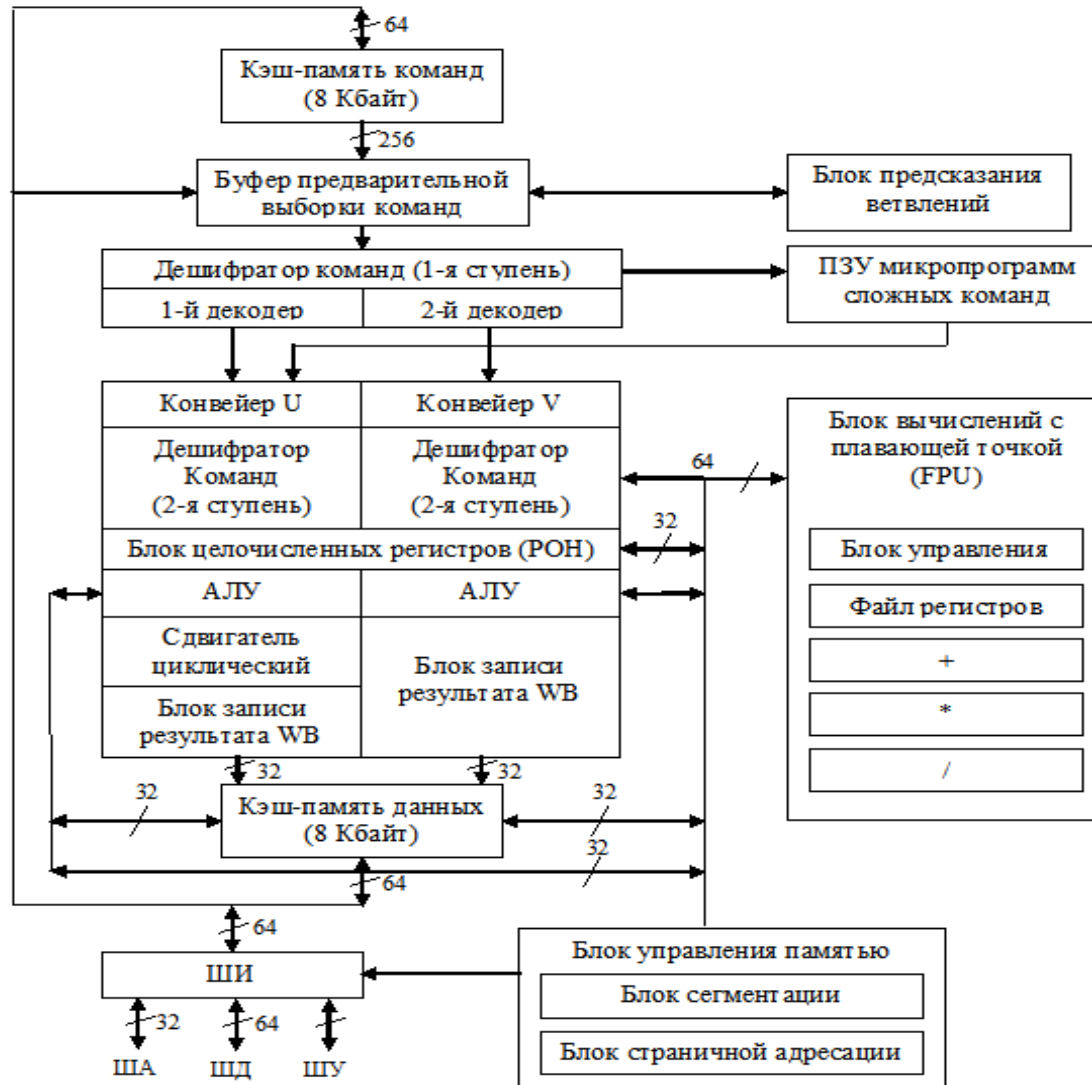
- В настоящее время доминирующее положение среди процессоров, используемых для персональных компьютеров, серверов, промышленных систем, занимают CISC-процессоры Pentium фирмы «Intel» и их клоны с системой команд x86 (более 80 %). Архитектура и тенденции развития этих процессоров отражают достижения всей компьютерной индустрии.
- Остальную долю занимают процессоры (большинство из них являются RISC-процессорами) других семейств, например, SPARC компании «SUN», Alpha компании «DEC» и др.
- Особенности организации и функционирования современных процессоров рассмотрим на примере процессоров Pentium фирмы «Intel».

Архитектурные особенности процессоров Pentium

Микропроцессор Pentium (1993 год) является первым суперскалярным процессором для персональных компьютеров. Включает в себя следующие функциональные блоки:

- 1) блок ШИ;
- 2) два 5-ступенчатых конвейера (U и V) выполнения команд целочисленных вычислений;
- 3) отдельные кэши команд и данных уровня $L1$ объёмом 8 Кбайт каждый;
- 4) блок вычислений с плавающей точкой FPU (Float Point Unit), организованный в виде конвейера;
- 5) блок предсказания ветвлений;
- 6) блок управления памятью.

Архитектурные особенности процессоров Pentium



Функциональная
схема процессора
Pentium

Архитектурные особенности процессоров Pentium

Большинство команд выполняется за один такт. В случае выполнения сложных команд используется расширенный микрокод из ПЗУ микропрограмм сложных команд.

Конвейеры Pentium реализуют традиционные пять этапов выполнения команд (выборку, декодирование операции, чтение операндов, записи результатов).

При вычислении операций с плавающей точкой добавляются ещё три шага:

X1 – преобразование данных в формате расширенной сложности,

X2 – выполнение FPU-команды,

WF – округление результата и его запись в регистровый файл FPU.

Конвейер *U* может выполнять как целочисленные, так и команды с плавающей точкой. При этом команды арифметики с плавающей точкой не могут запускаться в паре с целочисленными командами. Кроме того, конвейер *U* содержит многоразрядный сдвигатель, используемый при выполнении арифметических, логических, циклических сдвигов, операций умножения и деления.

Конвейер *V* выполняет только целочисленные команды.

Архитектурные особенности процессоров Pentium

Использование независимых кэшей обеспечивает одновременный бесконфликтный доступ к ним. Кэш-память команд связана с буфером предварительной выборки 256-битовой шиной. Если выбираемая команда в кэш-памяти отсутствует, то выполняется чтение искомой команды из основной памяти и её загрузка в буфер предвыборки с одновременной записью в кэш команд.

Кэш-память данных соединяется с конвейерами U и V при помощи двух 32-разрядных шин, что обеспечивает возможность одновременного обращения к ней со стороны каждого конвейера.

Архитектурные особенности процессоров Pentium

Блок управления памятью осуществляет двухступенчатое формирование физического адреса ячейки памяти сначала в пределах сегмента, а потом в пределах страницы. Для управления работой внешнего кэша используются специальные аппаратные средства.

Архитектурным нововведением Pentium является специальный режим системного управления (SMM – System Management Mode), который разработан для перевода системы в состояние пониженного энергопотребления. Этот режим недоступен приложениям и управляется программой из ПЗУ на кристалле процессора. В режиме SMM Pentium использует иное, изолированное от других режимов пространство памяти.

Архитектурные особенности процессоров Pentium

Блок предсказания ветвлений формирует прогнозируемый адрес ветвления, и команды выбранной ветви загружаются в буфер предварительной выборки команд. Для предсказания ветвлений используются *статические* и *динамические* методы.

Статическим предсказанием перехода называется алгоритм предсказания, при котором прогнозируемое ветвление выбирается разработчиком процессора на основании тех или иных соображений и который не изменяется во время работы программы. Например, можно предположить, что все условные переходы «назад» будут выполняться, а переходы «вперёд» – нет. Такое предположение неплохо работает при реализации циклов, в которых команда условного перехода чаще всего помещается в конце тела цикла, и переход назад выполняется достаточно часто.

Архитектурные особенности процессоров Pentium

Динамические методы предсказания ветвлений характеризуются тем, что прогнозируемое направление перехода зависит от результатов предшествующего выполнения переходов и может меняться в ходе выполнения программы. Динамические методы базируются на предыстории выполнения команды условного перехода: для каждого условного перехода накапливается статистика поведения, и переход реализуется в соответствии с этой статистикой.

Динамические методы, по сравнению со статическими методами, являются более точными и эффективными. Вероятность правильного предсказания переходов в современных процессорах, в том числе, и в Pentium, достигает 87 %.

Важно отметить, что неправильное предсказание не приводит к ошибочному выполнению программы, оно только вызывает задержку в её выполнении.

Архитектурные особенности процессоров Pentium

Блок ШИ обеспечивает связь процессора с другими устройствами через системную шину, включающую в себя 64-разрядную шину данных и 32-разрядную шину адреса и шину управления. Процессор Pentium поддерживает работу систем с физической памятью до 4 Гбайт.

Развитие процессоров Pentium связано, прежде всего, с усовершенствованием технологии их производства, расширением системы команд, развитием алгоритмов предсказания переходов, методов загрузки конвейеров, способов взаимодействия с кэш-памятью.

Архитектурные особенности процессоров Pentium

Отличительной особенностью процессоров Pentium 6-го (Pentium Pro, Pentium II, Pentium III) и 7-го (Pentium IV) поколений от ранних моделей Pentium является то, что в них используется один конвейер с большим количеством исполнительных блоков. Например, обработка команд в Pentium IV осуществляется 20-ступенчатым конвейером, который условно можно разделить на три относительно независимых конвейера:

1. Входной конвейер упорядоченной обработки, который обеспечивает выборку команд из памяти, декодирование их во внутренние RISC-команды и устранение ложных взаимосвязей по данным и ресурсам.

2. Конвейер неупорядоченной обработки, собственно реализующий исполнение команд. При неупорядоченной обработке происходит более интенсивная загрузка конвейеров суперскалярного процессора. Чтобы гарантировать правильное исполнение программы, результаты команд, выполненных вне очереди, должны записываться по целевым адресам в том же порядке, в каком они следуют в исходной программе. Эту работу осуществляет специальный блок процессора – блок временного хранения результатов, выполненных вне очереди.

3. Конвейер вывода результатов исполнения команд, который осуществляет запись результатов в архитектурные регистры процессора и память в порядке, предусмотренном программой.

Архитектурные особенности процессоров Pentium

Эффективная работа гиперконвейера Pentium IV обеспечивается также за счёт спекулятивного исполнения команд, когда для непрерывной загрузки конвейера выполняются действия, не предусмотренные программным кодом.

В блоке предсказания ветвлений используется более совершенный, по сравнению с более ранними моделями процессоров Pentium, алгоритм предсказания ветвлений. Адреса команд переходов и метки ветвлений с подробной предысторией сохраняются в буфере (объёмом 4 Кбайта) блока предсказания ветвлений.

Ещё одним важным отличием процессора Pentium IV от ранних моделей является использование в его структуре вместо кэша команд кэша трасс.

Трассы – это последовательности микрокоманд, в которые декодированы команды x86, принадлежащие одной или нескольким ветвям исходной программы. В кэше могут размещаться до 12 Кбайт микрокоманд. В кэш трасс не попадают команды, которые никогда не будут использоваться. Кэш трасс совместно с блоком выборки образуют устройство предварительной обработки.

Программная модель процессоров Pentium

Объединяющей характеристикой рассмотренных микропроцессоров Pentium является то, что все они, несмотря на существенные различия в архитектуре, имеют одинаковую программную модель. Совокупность всех программно доступных регистров процессора образует его **программную модель**, показывающую ресурсы процессора, которыми может пользоваться программист.

Программная модель подразделяется на прикладную и системную. В состав *прикладной программной модели* (ППМ) процессора входят полный набор регистров, которые доступны прикладным программистам; особенности организации памяти и доступные способы адресации; типы данных и команд.

Системная программная модель (СПМ) процессора объединяет его программно доступные системные ресурсы, с помощью которых обеспечивается доступ к встроенным механизмам защиты и многозадачности. СПМ процессора в основном используется системными программистами.

Программная модель процессоров Pentium

Старшие модели процессоров Pentium программно совместимы с младшими моделями. Регистры ППМ удобно представить в виде четырёх групп 32-разрядных регистров:

1. РОН, которые могут использоваться в формате двойного слова (32 бита), слова (16 бит) и байта (8 бит): EAX (AX, AH, AL), EBX (BX, BH, BL), ECX (CX, CH, CL), EDX (DX, DH, DL).

2. Регистры указатели ESP (SP), EBP (BP) и индексные регистры ESI (SI), EDI (DI), которые доступны в формате двойного слова и слова.

3. Указатель команд EIP (IP).

4. Регистр флагов FLAGS или PSW.

Функции, выполняемые регистрами, в подавляющем большинстве аналогичны функциям рассмотренных ранее регистров.

Регистр FLAGS, помимо уже известных флагов состояния и управления, содержит системные флаги, которые управляют обработкой исключений, вводом-выводом и рядом других функций. Большинство системных флагов предназначено для использования операционной системой.

Программная модель процессоров Pentium

СПМ включает в себя следующий набор регистров:

- 1) 6-байтовые регистры GDTR (Global Descriptor Table Register) и IDTR (Interrupt Descriptor Table Register) адресов глобальной дескрипторной таблицы (GDT) и таблицы прерываний (IDT);
- 2) 16-битный регистр LDTR (Local Descriptor Table Register), содержащий селектор сегмента локальной дескрипторной таблицы (LDT), и 16-битный регистр TR (Task Register), содержащий селектор сегмента состояния задачи TSS (Task State Segment);
- 3) 16-битные сегментные регистры CS, SS, DS, ES, FS, GS, содержащие селекторы дескрипторов текущих сегментов кода, стека и данных;
- 4) 32-битные регистры управления процессором CR (Control Register): CR0 - CR4: хранят признаки состояния процессора, общие для всех задач;
- 5) 8-байтовые программно недоступные кэш-регистры для хранения текущих дескрипторов сегментов: стека – регистр SSt, кода – регистр CSt, данных – регистры DSt, ESt, FSt, GSt, а также сегментов LDT и TSS;
- 6) 32-разрядные регистры отладки DR (Debug Register): DR0 – DR7, предназначенные для аппаратной отладки программ;
- 7) 32-разрядные регистры тестирования TR (Test Register): TR1 – TR12, используемые для тестирования внутренних устройств процессора.

Программная модель процессоров Pentium

Начиная с Pentium 286 – возможность работы в защищённом режиме, что обеспечивало аппаратную поддержку многозадачности. Защищённый режим работы процессора Pentium сохранил сегментную модель памяти. Однако сегмент в защищённом режиме – это объект, имеющий строго определённый размер и обладающий набором атрибутов, по которым выполняется аппаратная защита памяти со стороны процессора.

Каждый сегмент имеет свой **дескриптор** (описатель сегмента). Дескрипторы сегментов хранятся в специальных системных сегментах – **дескрипторных таблицах**. Существует три типа дескрипторных таблиц: глобальная GDT (одна в системе), локальная LDT (своя для каждой задачи) и таблица прерываний IDT. Каждый элемент (дескриптор) таблицы описывает свой сегмент памяти. Сегменты памяти не пересекаются. Адреса начала GDT и IDT хранятся в специальных программно доступных регистрах процессора GDTR и IDTR соответственно.

Регистры LDTR и TR содержат **селекторы** (индексы дескрипторов), с помощью которых из таблицы GDT выбираются дескрипторы с базовыми адресами LDT и TSS отдельных задач.

Программная модель процессоров Pentium

Система команд процессоров Pentium поддерживает типы команд, рассмотренные ранее. Набор команд обеспечивает выполнение операций над 8-, 16- и 32-разрядными операндами; содержит безадресные, 1-адресные и 2-адресные команды.

Система команд процессоров Pentium обеспечивает поддержку 11 режимов адресации: 9 рассмотренных ранее и 2 дополнительных: базовый индексный с масштабированием и относительный базовый индексный с масштабированием. Вычисление эффективного адреса (32-разрядного внутрисегментного смещения) выполняется по формулам , приведённым ниже.

$$EA = \{ \text{Содержимое базовых регистров} \} \\ + \{ \text{Содержимое индексных регистров} \} \times \{ \text{Масштаб} \};$$

$$EA = \{ \text{Содержимое базовых регистров} \} \\ + \{ \text{Содержимое индексных регистров} \} \times \{ \text{Масштаб} \} \\ + \{ \text{Смещение } disp \}.$$

Двухбитное поле *Масштаб* (*Scale*) кодирует значение масштабного множителя (1, 2, 4, 8), который указывает размер элементов массива при использовании индексной адресации.

Аппаратная организация защиты в процессорах Pentium

Современные ВМ поддерживают многопрограммный (мультипрограммный) и многозадачный (мультизадачный) режимы работы.

Многопрограммный режим предполагает одновременное выполнение процессором нескольких программ, переключаясь между ними. Естественно, в каждый конкретный момент времени процессор может выполнять команды только определённой программы.

Многозадачностью называется способ организации работы компьютера, при котором в его оперативной памяти содержатся программы и данные для одновременного выполнения нескольких процессов обработки информации (задач).

Один из способов реализации многозадачности, называемый разделением времени, заключается в предоставлении каждой задаче некоторого интервала времени, в течение которого процессор выполняет команды соответствующей программы.

Если по истечении выделенного кванта времени обработка программы не заканчивается, она прерывается и становится в очередь программ, ожидающих обработки.

Аппаратная организация защиты в процессорах Pentium

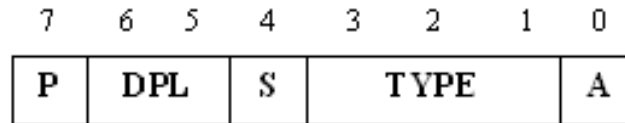
При реализации многозадачности необходимо обеспечить взаимную защиту программ и данных, относящихся к различным задачам.

Основной единицей защиты является сегмент. При каждом обращении к сегменту проверяется ряд параметров защиты, задаваемых в дескрипторе, в т. ч. предел, тип сегмента, права, уровень привилегий.

Специальное поле *предел* определяет границы сегмента. С его помощью процессор обнаруживает такие распространённые ошибки, как переполнение стека, неверные указатели и индексы массивов, неправильные адреса вызовов и переходов. Ошибки нарушения границы обнаруживаются в момент их возникновения, что существенно облегчает поиск их причины.

Аппаратная организация защиты в процессорах Pentium

Остальные параметры защиты хранятся в байте доступа (Access Rights) дескриптора, структура которого представлена на рисунке.



Специальный бит присутствия *P* указывает, местонахождение сегмента – в основной памяти ВМ ($P = 1$) или на диске ($P = 0$).

Бит доступа *A* фиксирует каждое обращение к сегменту.

Бит *S* кодирует назначение и способ использования дескриптора. При $S = 0$ дескрипторы являются системными: используются для обращения к системным сегментам TSS, LDT или являются шлюзами для задания разрешённых точек входа в защищённые программы ОС. Дескрипторы несистемных сегментов имеют бит $S = 1$.

Поле *TYPE* доопределяет тип дескриптора. Проверка типа позволяет обнаруживать ошибки при использовании сегмента не по назначению. Например, в сегмент кода нельзя записывать.

Поле привилегий *DPL* (*Descriptor Privilege Level*) задаёт разрешённый уровень привилегий для доступа к сегменту. Привилегии устанавливаются программами ОС в байте доступа дескриптора сегмента при заполнении дескрипторных таблиц.

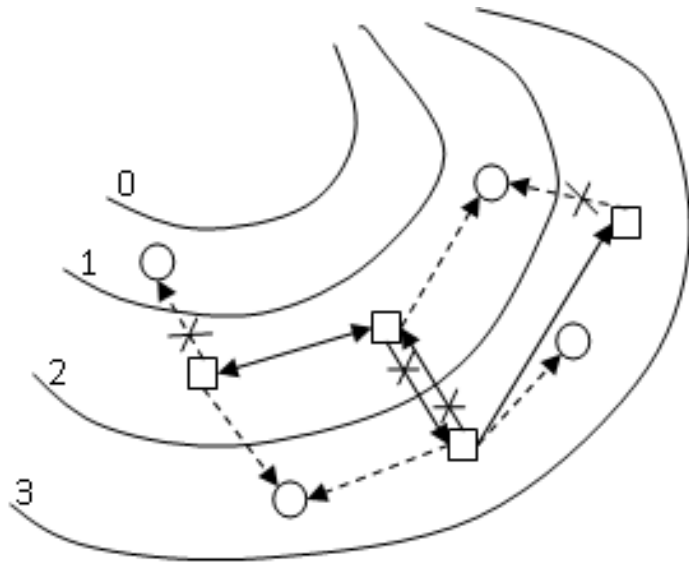
Аппаратная организация защиты в процессорах Pentium

Уровни привилегий (PL) удобно представлять в виде колец защиты: уровень привилегий $PL = 0$ отводится для ядра ОС, $PL = 1$ – утилитам ОС, $PL = 2$ – служебным программам ОС, системам управления базами данных и др., $PL = 3$ – пользовательским программам.

При любом межсегментном взаимодействии всегда участвуют два сегмента – текущий сегмент кода и сегмент, к которому происходит обращение (сегмент назначения).

Обратиться к данным сегмента назначения можно в том случае, если уровень его привилегий равен или меньше уровня привилегий текущего сегмента кода. При межсегментных передачах управления (доступе к другому сегменту кода) переход возможен, если уровень привилегий сегмента назначения равен уровню привилегий текущего сегмента кода (т.е., сегменты кода находятся в пределах одного кольца защиты).

Аппаратная организация защиты в процессорах Pentium



0, 1, 2, 3 – уровни защиты;

○ – данные;

□ – код;

→ (*) – разрешённый (запрещённый) переход при межсегментных передачах управления;

→ (*) – разрешённый (запрещённый) переход при доступе к данным.

Обращения в пределах колец защиты и между ними

Аппаратная организация защиты в процессорах Pentium

Перед тем как обратиться к операнду в памяти, программа загружает селектор в один из сегментных регистров, и аппаратным путём выполняется проверка возможности к адресуемому операнду.

Сначала процессор сравнивает формируемое внутрисегментное смещение со значением поля предела сегмента и определяет правильность адреса.

Затем процессор выполняет проверку привилегий сегмента, к которому происходит обращение (такая проверка зависит от типа сегмента назначения).

Непосредственно перед каждым обращением к операндам дополнительно аппаратным путём проверяется ещё ряд параметров защиты. Нарушение любого из них приводит к обработке особого случая и предотвращает обращение к операнду.

Передача управления в защищённом режиме осуществляется командами CALL, JMP, INT, RET, IRET, а также механизмами исключений и прерываний. При выполнении внутрисегментного перехода осуществляется только проверка границы перехода. При исполнении команд межсегментного перехода выполняется проверка привилегированности кодового сегмента назначения. Если межсегментный переход осуществляется в пределах одного кольца защиты, то доступ к вызываемому сегменту разрешён.

Аппаратная организация защиты в процессорах Pentium

Реализуя защиту, процессор запрещает пользовательским программам выполнять операции, осуществляемые программами ОС. Однако в любой системе есть некоторые стандартные процедуры, которые желательно реализовать на любом уровне привилегий, например, доступ пользовательских программ к утилитам ОС (они имеют $PL = 1$). Это приводит к наличию исключений в общих правилах защиты.

Процессоры Pentium обеспечивают два способа реализации межсегментных передач управления из менее привилегированного сегмента в более привилегированный: *обращение к согласованному сегменту и использование шлюзов вызова.*

Согласованный сегмент – это сегмент, у которого отсутствует защита по привилегиям. Выделение согласованного сегмента осуществляется с помощью специального бита подчинения *C* в поле *TYPE* байта доступа сегмента кода. Процедуры, размещённые в согласованном сегменте кода, можно вызывать из программ с любым уровнем привилегий. Наличие согласованных сегментов удобно для организации системных библиотек, программы которых должны быть доступны всем задачам.

Аппаратная организация защиты в процессорах Pentium

Шлюзы вызова – это специальные дескрипторы, которые определяют разрешённые точки входа в защищённые процедуры более высокого уровня. Благодаря шлюзам вызова, пользовательские программы получают от ОС только те обслуживания, которые ОС разрешает оказывать пользовательским программам. Шлюзы вызова удобно размещать в дескрипторных таблицах. Вызывающая программа обращается к шлюзу вызова, а он определяет точку входа в вызываемой процедуре.

32-разрядный дескриптор шлюза вызова имеет следующий формат: байт доступа, счётчик слов (1 байт), селектор сегмента назначения (1 байт) и смещение (1 байт).

Селектор сегмента назначения содержит адрес дескриптора кодового сегмента назначения. *Селектор* и *смещение* задают полный адрес перехода, который определяет вход в защищённую процедуру только в разрешённой точке.

При передаче управления более защищённой программе может потребоваться копирование параметров из стека вызывающей программы в стек вызываемой. Для этого в шлюзе вызова предусмотрено поле – *счётчик слов*, в котором указывается число копируемых параметров.

Доступ к шлюзу вызова контролируется механизмом защиты. Селекторы шлюзов вызова можно загружать только в сегментный регистр кода CS. При доступе к шлюзу действуют такие же ограничения, как и при обращении к данным .

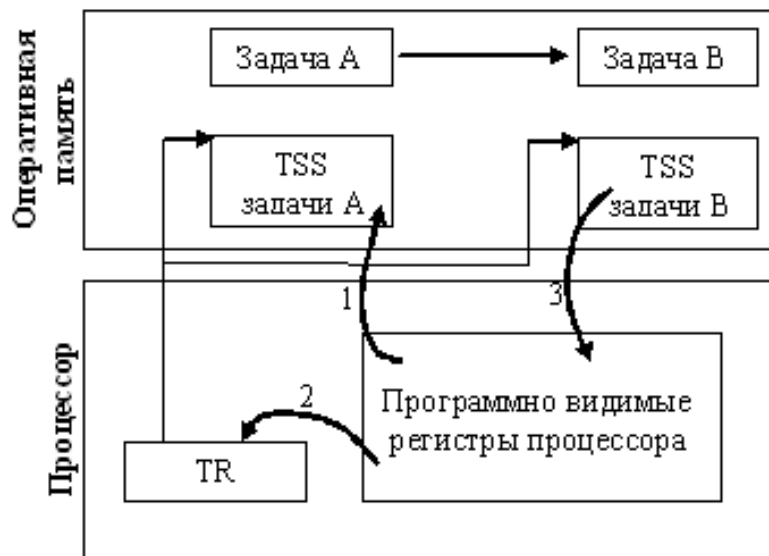
Аппаратные средства поддержки многозадачности

В структуре процессоров Pentium имеются средства поддержки многозадачного режима, с помощью которых реализуется защита и быстрое переключение задач: специальная структура данных, организованная в виде сегмента состояния задачи TSS, дескриптор сегмента TSS, дескриптор шлюза задачи, 16-разрядный регистр TR (Task Register) и связанный с ним программно недоступный теневой 64-разрядный регистр TRt.

Переключение задачи сходно с вызовом процедуры, но требует сохранения большего количества информации о состоянии процессора. Содержимое регистров процессора, участвующих в выполнении задачи, называется ***средой задачи***.

Для каждой задачи создаётся собственный сегмент TSS, который фактически характеризует состояние виртуального процессора задачи в многозадачной системе.

Аппаратные средства поддержки многозадачности



*Схема переключения
задач*

Выполнение задач процессора осуществляется в соответствии с планом и в порядке, определяемом ОС. При переключении задач среда выполняемой задачи переписывается в сегмент TSS этой задачи (на этот сегмент указывает регистр текущей задачи TRt) – (стрелка 1). После этого в регистр TR помещается селектор нового сегмента TSS (стрелка 2), и в регистры процессора из сегмента TSS вызываемой задачи загружается содержимое её среды (стрелка 3).

Выполнение программы новой задачи продолжается с команды, адрес которой указан в регистре EIP новой задачи. Указанные операции реализуются автоматически, с помощью аппаратных средств процессора. При этом в стеке никакой информации не сохраняется, управление передаётся полностью в среду выполняемой задачи.

Аппаратные средства поддержки МНОГОЗАДАЧНОСТИ

Основным назначением механизма переключения задач является организация очередных переходов между выполняемыми программами. Переключение программ может производиться командами JMP и CALL типа FAR (межсегментные переходы), командами вызова прерываний (INT n) или командой IRET, а также аппаратными прерываниями и ловушками.

При выполнении межсегментных переходов в защищённом режиме в регистре EIP содержится смещение (адрес команды внутри сегмента), а в регистре CS может находиться адрес либо дескриптора сегмента, либо шлюза дескриптора сегмента, а также дескриптора TSS или дескриптора шлюза TSS.

Сегмент TSS определяется одноимённым сегментным дескриптором, который может находиться только в глобальной дескрипторной таблице GDT. Этот дескриптор, помимо адреса, содержит ещё указание размера и уровня привилегий сегмента TSS. Выбор дескриптора TSS текущей задачи в таблице GDT осуществляется с помощью селектора, загружаемого в регистр TR.

С сегментом TSS нельзя явно проводить операции записи, чтения и выполнения. Для этого используются дополнительные сегменты данных с разрешённой записью.

Перспективы развития процессоров

Совершенствование технологии производства ИС приводит к росту уровня интеграции, уменьшению задержек в вентилях и связях, снижению энергопотребления при переключении вентиля.

С ростом уровня интеграции увеличиваются ресурсы на кристалле и повышается тактовая частота работы микросхем, что позволяет повышать производительность процессоров.

Первое направление связано с увеличением объёма внутренней кэш-памяти и совершенствованием способов её организации.

Второе – с реализацией в процессорах принципов конвейеризации и параллельной обработки в нескольких конвейерах на разных стадиях выборки и выполнения команд.

Практически все накопленные в процессе конкуренции различных фирм архитектурные решения находят своё воплощение в новых архитектурах. По этой причине в архитектуре современных процессоров различных производителей много общего, и ставится вопрос об унификации архитектур.

Перспективы развития процессоров

На сегодняшний день можно выделить следующие основные тенденции развития архитектур современных процессоров:

1. Применение суперскалярной обработки с динамическим параллелизмом в процессорах с чисто аппаратным механизмом выборки несвязанных команд программы из памяти и параллельном запуске их на исполнение (например, процессоры Pentium, PowerPC, Alpha, SPARC и др.).

2. Развитие суперскалярной обработки со статическим параллелизмом, при котором выявление скрытого параллелизма и определение возможности параллельного исполнения команд возлагается на оптимизирующий компилятор (например, процессоры Itanium, Crusoe и др.).

3. Использование аппаратных средств, обеспечивающих векторную обработку данных.

Перспективы развития процессоров

Современный процессор – это 64-разрядный суперконвейерный, суперскалярный процессор с RISC-операционным ядром и большим числом дополнительных блоков, реализующий динамическое исполнение команд.

Для эффективной обработки данных мультимедиа и графики система команд современных процессоров расширяется за счёт специализированных команд мультимедийной обработки: например, команд расширений MMX, SSE, SSE2 в процессорах компании «Intel»; ActiVec – «IBM», «Motorola»; VIS – «SPARC»; 3DNow! – «AMD».

Для унификации структур обработки данных в структуры некоторых современных процессоров включают специальные преобразователи исходных кодов команд во внутренние машинные команды «исполнительного процессора» (например, сложных CISC-команд в RISC-команды в процессорах Pentium, AMD или VLIW-команды в процессоре Crusoe).

Масштабные исследования по созданию процессорных элементов и компьютеров в целом ведутся с использованием принципиально иной элементной базы: биполярных молекул, молекул ДНК, квантовых кубитов и света.