НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ УКРАИНЫ

“КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ”

ФАКУЛЬТЕТ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Реферат

по архитектуре компьютеров

на тему: «**Виды и принципы работы кэш-памяти**»

Выполнил: студент 3-го курса

ФИВТ, группы ИВ-01

Ромас А.С.

Зачетная книжка №0123

Киев 2012

Содержание:

[**1. Общие понятие о кэшировании и кэш-памяти** 3](#_Toc344078967)

[**2. Внутренний кэш процессора** 5](#_Toc344078968)

[**3. Внешний кэш процессора** 8](#_Toc344078969)

[**4. Дополнительная кэш-память и программное кэширование** 10](#_Toc344078970)

[**4.1. Кэширование дисков** 10](#_Toc344078971)

[**4.2. Файл подкачки как кэш** 11](#_Toc344078972)

[**5. Уровни кэша** 12](#_Toc344078973)

[**6. Кэш-память микропроцессоров Pentium III** 13](#_Toc344078974)

[**Katmai** 14](#_Toc344078975)

[**Coppermine** 14](#_Toc344078976)

[**Tualatin** 15](#_Toc344078977)

[**7. Устройство кэш-памяти и стратегии кэширования** 15](#_Toc344078978)

[**Заключение** 21](#_Toc344078979)

[**Список литературы** 22](#_Toc344078980)

Одним из важнейших устройств компьютера является память, или запоминающее устройство (ОЗУ). По определению, данном в книге "Информатика в понятиях и терминах", ОЗУ - "функциональная часть цифровой вычислительной машины, предназначенной для записи, хранения и выдачи информации, представленных в цифровом виде." Однако под это определение попадает как собственно память, так и внешние запоминающие устройства (типа накопителей на жестких и гибких дисках, магнитной ленты, CD-ROM), которые лучше отнести к устройствам ввода/вывода информации. Таким образом под компьютерной памятью в дальнейшем будет пониматься только "внутренняя память компьютера: ОЗУ, ПЗУ, кэш память и флэш-память".

Однако ОЗУ работает намного медленнее процессора. Другое дело - кэш. Очень быстрая и очень дорогая память. Но именно из-за дороговизны в компьютерной системе объем кэш составляет всего 10-15% от объема обычной ОЗУ. А также используются специальные алгоритмы работы материнской платы и процессора, обеспечивающие максимальную производительность при наличии встроенной кэш-памяти.

**1. Общие понятие о кэшировании и кэш-памяти**

Существующая уже более 100 лет архитектура компьютера, предложенная еще Фон Нейманом, практически не претерпела изменения. Все также процессор окружает оперативная, постоянная и внешняя память. И все также оперативная память, являющаяся одной из скоростных элементов памяти внутри компьютера, не достигает по скорости процессор на несколько порядков.

Естественно, что такой «тормоз» в виде одного из необходимых элементов компьютера не устраивал его создателей. Тем более, что в больших компьютерах уже давно существовало понятие кэш-памяти (cashe-memory), более быстрой, чем ОЗУ, но меньшей емкости. Было решено перенести ее архитектуру с БЭВМ на обычные персональные компьютеры.

Принцип работы кэш-памяти заключается в следующем. Процессор редко использует весь объем ОЗУ практически одновременно. Скакать из одного угла памяти в другой, периодически пошвыриваясь по всему ее объему – это не лучший способ использования ресурсов компьютера. Зачастую все обращения процессора к памяти сосредоточены в небольшой области (как показывает статистика – 5-10% от общего объема). Если данные из этой области как либо аппаратно скопировать в кэш, а затем постоянно сверять кэш и ОЗУ на предмет целостности данных, то можно обеспечить режим работы, при котором процессор будет обращаться только к кэш-памяти, тратя на это значительно меньше ресурсов и времени, чем обычно.

Естественно, что весь объем ОЗУ скопировать в кэш нет возможности – такой объем кэш-памяти по цене сравнивается со стоимостью компьютера, а смысла уменьшать объем ОЗУ тоже нет. Было решено реализовать алгоритм работы процессора, кэш-памяти и ОЗУ аппаратно, чтобы не тратить ресурсы процессора.

Принцип заключается в следующем: когда процессор обращается к определенной ячейке памяти, сегмент памяти определенного объема (этот объем называется объемом страницы кэш) копируется в кэш полностью. Если процессор дальше не совершит глобальный скачек на другой, далекий от текущего, адрес памяти, то дальнейшая работа процессора будет происходить напрямую с кэш, минуя ОЗУ, а контроллер кэш-памяти в промежутках, когда процессор занят вычислениями (либо параллельно с работой процессора) будет восстанавливать верные данные в ОЗУ либо в кэш (в случае наличия устройств, напрямую работающих с памятью). Естественно, чем больше будет страниц и чем больше будет их объем – тем выше будет скорость работы процессора.

Начиная с 386-х процессоров кэш-память устанавливалась на большинство материнских плат. Все это позволяло повышать производительность систем не в разы, а на порядки. Но с выходом 486 процессора кэш-память была разделена: появился кэш, реализованный непосредственно на кристалле процессора (в 486DX). Несмотря на малый его объем – 1 Кб – это уже был скачек в производительности и очень большой: ведь кэш, расположенный на кристалле процессора работает на частоте процессора, что в несколько раз превышает скорость работы кэш-памяти, установленной на материнской плате.

Начиная с 486 процессоров кэш-память испытывает подобное разделение на кэш 1-го уровня – установленный на кристалле (или в одном корпусе) процессора и работающий с ним на одной частоте, и кэш 2-го уровня, установленный на материнской плате, работающий быстрее обычной ОЗУ, но медленнее КЭШа первого уровня.

Однако этим не ограничивается использования кэш-памяти. Ведь таким образом ее можно использовать там, где существует проблема быстродействия, но есть возможность упорядочить данные. К таким применениям относят:

аппаратное кэширование жестких дисков (кэш-память устанавливается непосредственно на жестком диске либо на специальном контроллере);

программное кэширование CD-ROM, а также прочих устройств хранения информации (программно – при помощи операционной системы, аппаратно – на самом устройстве либо на контроллере).

И не только: сегодня зачастую даже самое простейшее устройство обладает своей памятью, работающей быстрее, чем само устройство. К таким относят принтеры, сканеры, модемы и т.д.

Но все-таки наиболее популярной является кэш-память первого уровня (процессорная) и второго (установленная на материнской плате).

**2. Внутренний кэш процессора**

Внутренне кэширование обращений к памяти применяется в процессорах, начиная с 486-го. С кэшированием связаны новые функции процессоров, биты регистров и внешние сигналы.

Процессоры 486 и Pentium имеют внутренний кэш первого уровня, в Pentium Pro и Pentium II имеется и вторичный кэш. Процессоры могут иметь как единый кэш инструкций и данных, так и общий. Выделенный кэш инструкций обычно используется только для чтения. Для внутреннего кэша обычно используется наборно-ассоциативная архитектура.

Строки в кэш-памяти выделяются только при чтении, политика записи первых процессоров 486 – только Write Through (сквозная запись) – полностью программно-прозрачная. Более поздние модификации 486-го и все старшие процессоры позволяют переключаться на политику Write Back (обратная запись).

Работу кэша рассмотрим на примере четырехканального наборно-ассоциативного кэша процессора 486. Кэш является несекторированным – каждый бит достоверности (Valid bit) относится к целой строке, так что стока не может являться “частично достоверной”.

Работу внутренней кэш-памяти характеризуют следующие процессы: обслуживание запросов процессора на обращение к памяти, выделение и замещение строк для кэширования областей физической памяти, обеспечение согласованности данных внутреннего кэша и оперативной памяти, управление кэшированием.

Любой внутренний запрос процессора на обращение к памяти направляется на внутренний кэш. Теги четырех строк набора, который обслуживает данный адрес, сравниваются со старшими битами запрошенного физического адреса. Если адресуемая область представлена в строке кэш-памяти (случая попадания –cache hit), запрос на чтение обслуживается только кэш-памятью, не выходя на внешнюю шину. Запрос на запись модифицирует данную строку, и в зависимости от политики записи либо сразу выходит на внешнюю шину (при сквозной записи), либо несколько позже (при использовании алгоритма обратной записи).

В случае промаха (Cache Miss) запрос на запись направляется только на внешнюю шину, а запрос на чтение обслуживается сложнее. Если этот зарос относится к кэшируемой области памяти, выполняется цикл заполнения целой строки кэша – все 16 байт (32 для Pentium) читаются из оперативной памяти и помещаются в одну из строк кэша, обслуживающего данный адрес. Если затребованные данные не укладываются в одной строке, заполняется и соседняя. Заполнение строки процессор старается выполнить самым быстрым способом – пакетным циклом с 32-битными передачами (64-битными для Pentium и старше).

Внутренний запрос процессора на данные удовлетворяется сразу, как только затребованные данные считываются из ОЗУ – заполнение строки до конца может происходить параллельно с обработкой полученных данных. Если в наборе, который обслуживает данный адрес памяти, имеется свободная строка (с нулевым битом достоверности), заполнена будет она и для нее установится бит достоверности. Если свободных строк в наборе нет, будет замещена строка, к которой дольше всех не было обращений. Выбор строки для замещения выполняется на основе анализа бит LRU (Least Recently Used) по алгоритму “псевдо-LRU”. Эти биты (по три на каждый из наборов) модифицируются при каждом обращении к строке данного набора (кэш-попадании или замещении).

Таким образом, выделение и замещение строк выполнятся только кэш-промахов чтения, при промахах записи заполнение строк не производится. Если затребованная область памяти присутствует в строке внутреннего кэша, то он обслужит этот запрос. Управлять кэшированием можно только на этапе заполнения строк; кроме того, существует возможность их аннулирования – объявления недостоверными и очистка всей кэш-памяти.

Очистка внутренней кэш-памяти при сквозной записи (обнуление бит достоверности всех строк) осуществляется внешним сигналом FLUSH# за один такт системной шины (и, конечно же, по сигналу RESET). Кроме того, имеются инструкции аннулирования INVD и WBINVD. Инструкция INVD аннулирует строки внутреннего кэша без выгрузки модифицированных строк, поэтому ее неосторожное использование при включенной политике обратной записи может привести к нарушению целостности данных в иерархической памяти. Инструкция WBINVD предварительно выгружает модифицированные строки в основную память (при сквозной записи ее действие совпадает с INVD). При обратной записи очистка кэша подразумевает и выгрузку всех модифицированных строк в основную память. Для этого, естественно, может потребоваться и значительное число тактов системной шины, необходимых для проведения всех операций записи.

Аннулирование строк выполняется внешними схемами – оно необходимо в системах, у которых в оперативную память запись может производить не только один процессор, а и другие контроллеры шины – процессор или периферийные контроллеры. В этом случае требуются специальные средства для поддержания согласованности данных во всех ступенях памяти – в первичной и вторичной кэш-памяти и динамического ОЗУ. Если внешний (по отношению к рассматриваемому процессору) контроллер выполняет запись в память, процессору должен быть подан сигнал AHOLD. По этому сигналу процессор немедленно отдает управление шиной адреса A[31:4], на которой внешним контроллером устанавливается адрес памяти, сопровождаемый стробом EADS#. Если адресованная память присутствует в первичном кэше, процессор аннулирует строку – сбрасывает бит достоверности этой строки (она освобождается). Аннулирование строки процессор выполняет в любом состоянии.

Управление заполнением кэша возможно и на аппаратном и на программном уровнях. Процессор позволяет кэшировать любую область физической памяти, но внешние схемы могут запрещать процессору кэшировать определенные области памяти. Это делается по различным причинам, зачастую связанным с определенными условиями создания компьютерной системы.

**3. Внешний кэш процессора**

В отличие от внутренней кэш-памяти, внешняя больше напоминает обычную память. Однако алгоритм работы с ней практически такой же.

Внешняя кэш-память состоит из памяти данных, построенная на микросхемах SRAM, и контроллера кэша. В кэш-памяти хранится информация, копируемая из основной оперативной памяти. Каждый раз при обращении микропроцессора к памяти контроллер кэш-памяти проверяет наличие данных в кэше. Если эти данные в кэше есть (“попадание”), то микропроцессор получает данные из кэша. Если этих данных нет (“промах”), выполняется обычный цикл обращения к оперативной памяти DRAM.

Основным фактором, определяющим вероятность попадания, является емкость кэш-памяти. Как правило, при объеме кэша в 2 Кбайта вероятность попадания составляет от 50 до 60%. Поскольку размер кэш-памяти на современных компьютерах превышает 256 Кбайт, то вероятность попадания будет выше 90% (для компьютеров с объемом памяти ~ 256 Мбайт.)

Для реализации кэш-памяти в настоящее время разработаны эффективные однокристальные контроллеры. Наиболее широкое распространение получили контроллеры i82385 фирмы Intel и A38152 фирмы Asustec Microsystems.

Контроллер i82385 поддерживает 32 Кбайта кэш-памяти, и может работать в двух конфигурациях:

кэш-память с прямым отображением;

двухканальная модульно-ассоциативная кэш-память.

Первая конфигурация характеризуется простотой реализации, однако она оказывается неэффективной при работе в мультизадачных системах. В двухканальной реализации кэш-память разбивает все 4 Гбайтное адресное пространство на 262144 страницы по 16 Кбайт. 32-х разрядный физический адрес состоит из четырнадцатиразрядного адреса, определяющего информацию в кэш-памяти, и восемнадцатиразрядного тега, определяющего номер страницы. Каждый адрес оперативной памяти может быть отображен в одну из двух ячеек кэш-памяти.

Особенность контроллера кэш-памяти – обеспечение возможности параллельной работы микропроцессора с кэш-памятью и периферийных устройств с оперативной памятью в режиме прямого доступа. При записи данных по адресам, находящихся в кэше, контроллер ликвидирует копии этих блоков в кэше. Всю работу по синхронизации данных в DRAM и кэше берет на себя этот контроллер.

Одним из популярных контроллеров кэш-памяти является однокристальный контроллер кэш-памяти фирмы ASUSTEC, совместно с памятью данных 32 Кбайта обеспечивает вероятность попадания более 95%. Это достигается благодаря использованию четырехканального модульно-ассоциативного обращения, который отображает адрес оперативной памяти в одну из четырех ячеек кэш-памяти. При этом, вследствие организации последовательного обращения к памяти данных, требуется подключение всего одного банка памяти данных.

Контроллер A38152 фирмы Asustec имеет аппаратные и программные средства, обеспечивающие связанность информации: логика слежения за шиной, которая обеспечивает ликвидацию копий блоков в кэш-памяти, задание области адресов, не отображаемой в кэш-память.

На многих материнских платах можно выбирать между одноуровневой или многоуровневой системами организации памяти. По умолчанию устанавливается ражим многоуровневой памяти. Если Вы установите режим одноуровневой памяти, то кэш-память SRAM просто добавляется к адресному пространству основной оперативной памяти. Одноуровневую память лучше использовать, когда внутренний кэш процессора по объему превосходит емкость кэш-памяти на материнской плате.

**4. Дополнительная кэш-память и программное кэширование**

**4.1. Кэширование дисков**

Однако кэш-память процессора и оперативной памяти – это не единственный вариант ее использования. Фактически кэширование как процесс увеличения скорости за счет подмены устройства более быстрым, используется уже давно. Причем не аппаратно, а программно (аппаратно тоже, но это остается за высоким забором закрытых технологий производителей жестких дисков и других устройств).

Операционная система MS-DOS с 4 версии имеет в своем составе программу smartdrive, позволяющую наиболее используемые данные с жесткого диска (или дисков) перенести в память компьютера. При этом ОС, обращаясь к диску, будет фактически обращаться к памяти, за счет чего увеличится быстродействие. Позже, при наименьшей загрузке данные из памяти будут скинуты на диск.

Естественно, так как идеала не существует, то и в данной модели приходится чем-то жертвовать. Такой жертвой является свободная память. Однако если на первых компьютерах с объемом памяти 1-16 Мб это было существенно, то сегодня, когда объем памяти нередко всего лишь в несколько раз меньше объема диска, выделить под кэш 4-16 Мб не так уж и страшно.

Но сегодня думать об этом пользователю нет необходимости. Современные операционные системы Windows 95/98/Me/2000/XP организуют кэш для жесткого диска (и вообще для всех устройств хранения информации) в памяти самостоятельно, автоматически выделяя под любое устройство требуемый объем и делая это совершенно незаметно для пользователя.

Единственным недостатком подобной организации остается то, что в случае пропадания питания (и не очень хорошем блоке питания) данные из памяти на диск ОС сбросить не успеет, и они будут утеряны.

Стоит добавить, что сегодня практически все жесткие диски имеют встроенный кэш для увеличения работы. Причем ему не свойственен недостаток, описанный нами выше. Современные технологии позволяют жесткому диску даже после пропадания питания нормально очищать кэш и завершать работу.

**4.2. Файл подкачки как кэш**

Однако говоря о кэш, нельзя не упомянуть о такой его своеобразной реализации, как файл подкачки в Windows. Казалось бы: если компьютер и так тормозит из-за работы с ОЗУ, то зачем создавать лишние проблемы и скидывать часть данных из ОЗУ на диск - ведь это еще сильнее уменьшит скорость работы системы. Однако цели здесь совсем иные.

Основное назначение файла подкачки заключается в том, чтобы позволить активной программе воспользоваться практически всем объемом памяти, установленным на компьютере. Естественно, что если вы работаете в word и печатаете (уже два часа) три страницы какого-то текста, то все 512 Мб ОЗУ вам не нужны. Но если сфера вашей деятельности – видео, звуки либо работа в интернете (особенно в роли web-мастера), то это уже является существенным плюсом ОС. Вы можете запустить все нужные вам программы, а затем не только пользоваться практически 100% объема ОЗУ, установленного на компьютере, но и использовать его в качестве этакого громадного буфера обмена.

Кроме того, данная организация позволяет выполнить еще одну функцию: периодическую оптимизацию данных, содержащихся в ОЗУ с целью более рационального его использования. Удаление данных, уже не используемых, а также сохранение данных, редко используемых, на диске, позволяет высвободить дополнительный объем пространства.

**5. Уровни кэша**

Кэш центрального процессора разделён на несколько уровней. В универсальном процессоре в настоящее время число уровней может достигать 3. Кэш-память Pentium III имеет два уровня. Самой быстрой памятью является кэш первого уровня — L1-cache. Небольшая (несколько десятков килобайт) сверхбыстрая память, предназначнная для хранения промежуточных результатов вычислений.  По сути, она является неотъемлемой частью процессора, поскольку расположена на одном с ним кристалле и входит в состав функциональных блоков. Большинство процессоров без L1 кэша не могут функционировать. L1 кэш работает на частоте процессора, и, в общем случае, обращение к нему может производиться каждый такт. Зачастую является возможным выполнять несколько операций чтения/записи одновременно. Латентность доступа обычно равна 2−4 тактам ядра. Объём обычно невелик — не более 128 Кбайт. Вторым по быстродействию является L2-cache — кэш второго уровня. Хотя эта память чуть помедленнее, зато больше. Обычно он расположен на кристалле, как и L1. Объём L2 кэша от 128 Кбайт до 1−12 Мбайт. В современных многоядерных процессорах кэш второго уровня, находясь на том же кристалле, является памятью раздельного пользования — при общем объёме кэша в nM Мбайт на каждое ядро приходится по nM/nC Мбайта, где nC количество ядер процессора. Обычно латентность L2 кэша, расположенного на кристалле ядра, составляет от 8 до 20 тактов ядра. Проблему синхронизации между различными кэшами (как одного, так и множества процессоров) решается Когерентностью кэша. Существует три варианта обмена информацией между кэш-памятью различных уровней, или, как говорят, кэш-архитектуры: инклюзивная, эксклюзивная и неэксклюзивная.  
Инклюзивная архитектура предполагает дублирование информации кэша верхнего уровня в нижнем (предпочитает фирма Intel). Эксклюзивная кэш-память предполагает уникальность информации, находящейся в различных уровнях кэша. В неэксклюзивной кэши могут вести себя как угодно. Одна из фундаментальных характеристик кэш-памяти — уровень ассоциативности — отображает её логическую сегментацию. Дело в том, что последовательный перебор всех строк кэша в поисках необходимых данных потребовал бы десятков тактов и свёл бы на нет весь выигрыш от использования встроенной в ЦП памяти. Поэтому ячейки ОЗУ жёстко привязываются к строкам кэш-памяти (в каждой строке могут быть данные из фиксированного набора адресов), что значительно сокращает время поиска. С каждой ячейкой ОЗУ может быть связано более одной строки кэш-памяти: например, n-канальная ассоциативность обозначает, что информация по некоторому адресу оперативной памяти может храниться в n местах кэш-памяти. При одинаковом объеме кэша схема с большей ассоциативностью будет наименее быстрой, но наиболее эффективной

**6. Кэш-память микропроцессоров Pentium III**

Процессор Intel Pentium III - процессор архитектуры P6, включает в себя: динамическое исполнение команд, системную шину с множественными транзакциями и технологию Intel MMX™ для обработки данных мультимедиа. Технология изготовления с разрешающей способностью 0.25 микрон позволяет разместить на кристалле более 9.5 миллионов транзисторов. Процессор содержит 32 Kб неблокируемой кэш-памяти первого уровня (16Кб/16Кб) и унифицированную неблокируемую кэш-память второго уровня емкостью 512 Кб, функционирующую на вдвое меньшей частоте, чем ядро. Процессор Intel® Pentium®III поддерживает кэширование памяти с объемом адресного пространства 4 Гб, и позволяет создавать масштабируемые системы с двумя процессорами и физической памятью объемом до 64 Гб.

## Katmai

Кэш второго уровня объёмом 512 Кб работает на половине частоты ядра и выполнен в виде двух микросхем BSRAM (производства Toshiba и NEC), расположенных друг над другом справа от кристалла процессора. В качестве tag-RAM используется микросхема Intel 82459AD, расположенная на обратной стороне процессорной платы под микросхемами кэш-памяти.

## Coppermine

Процессоры на ядре Coppermine выпускались по 180 нм технологии, имели интегрированную кэш-память второго уровня, работающую на частоте ядра. Кроме того, кэш-память имеет 256-битную шину (в отличие от процессоров на ядре Katmai, имевших 64-битную шину кэш-памяти), что значительно повышает её быстродействие. За счёт интегрированной кэш-памяти число транзисторов возросло до 28,1 млн  
  
Процессоры Pentium III 733 МГц на ядре Coppermine c уменьшенной до 128 Кб кэш-памятью второго уровня использовались компанией Microsoft в приставке Xbox. В отличие от процессоров Celeron на ядре Coppermine-128, также имеющих 128 Кб кэша, данные процессоры имеют 8-канальный ассоциативный кэш второго уровня (Celeron имеет 4-канальный ассоциативный кэш).

## Tualatin

Процессоры Pentium III-S на ядре Tualatin  имели 512 Кб кэш-памяти второго уровня и предназначались для высокопроизводительных рабочих станций и серверов. В процессорах Pentium III на ядре Tualatin 256 Кб кэш-памяти были аппаратно отключены. Частота системной шины составляла 133 МГц для обеих модификаций.

# 7. Устройство кэш-памяти и стратегии кэширования

Кэш-память является результатом попыток соединить достоинства быстрых SRAM и дешевых DRAM для создания максимально эффективной системы памяти. Принцип кэширования поясняется на рисунке 1.

|  |
| --- |
|  |
|  | 198.jpg |

Между процессором и основной памятью DRAM предусматривается быстрый кэш SRAM. В нем хранятся часто требуемые данные, которые он способен передавать очень быстро. Процесс управляется кэш-контроллером, который может обеспечивать различные режимы записи - такие, как  
  
сквозная или отложенная запись.  
  
  Рисунок 1. Принцип кэширования.  
Кэш-блок располагается между CPU и основной памятью; он состоит из кэш-контроллера и кэш-памяти SRAM. Они могут быть встроены в кристалл процессора (кэш-память, встроенная в кристалл), а могут существовать и в виде отдельного элемента.   
  
Поскольку последовательные операции доступа к памяти в основном обращаются к ограниченному пространству адресов, то имеет смысл разместить наиболее часто требуемые данные в небольшой быстродействующей памяти - кэш-памяти. Преимуществом такого подхода является существенное уменьшение времени доступа, которое при большом количестве операций доступа к памяти обеспечивает значительное повышение быстродействия. Данные и команды, которые в данный момент не требуются, могут храниться в более медленной основной памяти, что не приводит к заметному замедлению выполнения программы. Принцип кэширования, заключающийся в использовании небольшой SRAM и большой, но более медленной DRAM, сочетает в себе преимущества быстрых SRAM и более дешевых DRAM.



На рисунке 2 изображены основная и кэш память. Каждая строка — группа ячеек памяти содержит данные, организованные в кэш-линии. Размер каждой кэш-линии может различаться в разных процессорах, но для большинства x86-процессоров он составляет 64 байта. Размер кэш-линии обычно больше размера данных, к которому возможен доступ из одной машинной команды (типичные размеры от 1 до 16 байт). Каждая группа данных в памяти размером в 1 кэш-линию имеет порядковый номер. Для основной памяти этот номер является адресом памяти с отброшенными младшими битами. В кэше каждой кэш-линии дополнительно ставится в соответствие тег, который является адресом продублированных в этой кэш-линии данных в основной памяти.  
Рисунок 2. Диаграмма кэша памяти ЦПУ  
Когда процессор читает информацию, он обычно направляет соответствующий адрес в память. Когда процессору нужно выполнить команду, он сначала анализирует состояние своих регистров данных. Если необходимых данных в регистрах нет, он обращается к кэш-памяти первого уровня, а затем — к кэш-памяти второго уровня. Если данных нет ни в одной кэш-памяти, процессор обращается к оперативной памяти. И только в том случае, если нужных данных нет и там, он считывает данные с жесткого диска.  
  
Однако в нашем случае между процессором и адресом основной памяти находится кэш-контроллер. Он определяет, находятся ли нужные данные в кэш-памяти SRAM. Если да, то ситуация называется "кэш-попаданием". Случай, когда нужные данные находятся в основной памяти, называется "кэш-промахом". В первом случае кэш-контроллер читает данные из быстрой кэш-памяти и направляет их процессору. Это обычно происходит без ожидания, т.е. с максимальной скоростью шины. Запрос на чтение перехватывается кэш-памятью, и основная память о нем не знает.  
  
В случае кэш-промаха кэш-контроллер должен вначале прочитать данные из основной памяти; таким образом, запрос на чтение переадресуется в основную память. Поскольку это обычно занимает больше времени, то требуется определенное число циклов ожидания; кэш-контроллер сбрасывает сигнал готовности (или другой соответствующий сигнал), и процессор поэтому вставляет циклы ожидания. Одновременно с этим обращается к памяти и кэш-контроллер. Внутренняя организация большинства моделей кэш-памяти такова, что в том случае, когда происходит запрос на чтение информации, которой нет в кэш-памяти, из основной памяти в SRAM читаются не только непосредственно запрашиваемые байты данных, но и вся кэш-строка. Эта операция известна как заполнение кэш-строки. Перед тем, как записать в кэш-память новые данные, нужно, чтобы содержащиеся в ней результаты предыдущих операций были сохранены в основной памяти. Кэш-контроллер достаточно разумен для того, чтобы выполнить циклы чтения и записи данных в основную память независимо друг от друга. Байты данных, запрашиваемые процессором, немедленно, т.е. до завершения полной обработки всей кэш-строки, передаются ему кэш-контроллером.  
  
Кэш-строки обычно составляют 16 или 32 байта, поскольку, как уже говорилось ранее, данные и команды формируются в виде блоков; поэтому очень вероятно, что следующий запрос обратится к информации, содержащейся в той же кэш-строке. Это увеличивает скорость доступа. Кроме того, большинство кэш-контроллеров реализует так называемый пакетный режим, посредством которого считывается целиком блок данных, в котором содержится больше байт, чем позволяет ширина шины (поэтому для чтения блока данных требуется несколько циклов шины). Пакетный режим почти удваивает скорость передачи данных шиной, поэтому целая кэш-строка читается значительно быстрее, чем отдельное значение. Таким образом, организация кэш-памяти в виде кэш-строк увеличивает быстродействие системы.  
  
В случае, когда процессор записывает данные, кэш-контроллер определяет, находятся ли данные в кэш-памяти SRAM. Если да, то данные из процессора записываются в кэш. Существует три различных стратегии, определяющие дальнейшие действия кэш-контроллера: сквозная запись, отложенная запись (известная также как обратное копирование) и запись с размещением. Первые две стратегии относятся к случаю кэш-попадания, последняя - к ситуации кэш-промаха.  
  
Наиболее простым методом является сквозная запись, которая реализована в большинстве моделей кэш-памяти. Операция записи, инициированная процессором, всегда приводит к передаче данных в основную память, даже в случае кэш-попадания; все операции записи проходят через основную память. Это, естественно, касается также записи и обновления соответствующих ячеек кэш-памяти. Сквозная запись имеет тот недостаток, что все операции записи должны производиться также и с основной, более медленной памятью. Если не принять дополнительных мер, то это в принципе может привести к отключению кэш-памяти в режиме записи, результатом чего явится неоправданно большое время доступа при записи. По этой причине модели кэш-памяти со сквозной записью используют быстрые буферы записи, позволяющие буферизовать операции доступа на запись. Однако это возможно только до тех пор, пока буфер не заполнится. Следовательно, многократные запросы на запись неизбежно приводят к состоянию ожидания. С другой стороны, в мультипроцессорных системах стратегия сквозной записи автоматически обеспечивает согласованность содержимого основной памяти, поскольку все данные обновляются через основную память. Согласованность содержимого кэш-памяти, однако, в мультипроцессорных системах не гарантируется. Например, один из процессоров может переписать содержимое основной памяти, а кэш-память другого процессора может ничего об этом не знать. Только цикл просмотра может восстановить согласованность данных.  
  
Кэш с отложенной записью концентрирует все операции записи и обновляет только содержимое элемента кэш-памяти, не затрагивая содержимого основной памяти. Только по определенной команде измененное содержимое кэш-строки записывается в основную память, обновляя содержащуюся в ней информацию. В компьютерах Pentium это может быть выполнено программным путем, например, посредством команды WBIND (записать и аннулировать данные кэш-памяти), путем выдачи аппаратного сигнала FLUSH, при безусловной замене одной кэш-строки другой в случае отсутствия запрашиваемых данных в кэш-памяти, а также при выполнении внутреннего или внешнего цикла просмотра. Недостатком кэш-памяти с отложенной записью является то, что смена строк кэш-памяти занимает больше времени, поскольку до того, как новые данные могут быть записаны в кэш, прежнее содержимое кэш-памяти должно быть записано в основную память. Этот недостаток, однако, более чем компенсируется тем, что предшествующие запросы на запись не проходят через медленную общую память.  
  
Две стратегии кэш-памяти, описываемые здесь, не определяют ее поведения в случае кэш-промаха, т.е. если адрес, к которому происходит обращение, в кэш-памяти отсутствует. Если кэш реализует стратегию записи с размещением, кэш-контроллер заполняет пространство кэш-памяти кэш-строкой, в которую входят данные по запрашиваемому адресу. Обычно данные вначале записываются в основную память; затем кэш-контроллер считывает в кэш ту кэш-строку, элемент которой подлежит обновлению. Поскольку сначала происходит запись информации, процессор может немедленно возобновить выполнение программы. Кэш-контроллер независимо от этого выполняет запись с размещением параллельно с работой процессора. В наихудшем случае он вначале записывает обновленную кэш-строку в основную память, чтобы затем заполнить ее новыми данными. По этой причине и из-за сложности алгоритма работы кэш-памяти большинство моделей кэш-памяти не использует стратегию записи с размещением. Запросы на запись в случае кэш-промаха просто игнорируются кэш-памятью и передаются в основную память.  
  
Если к основной памяти имеют доступ другие процессоры или компоненты системы, как происходит, например, при работе DMA-контроллера, и содержимое памяти изменяется, кэш-контроллер должен информировать соответствующую SRAM о том, что данные в основной памяти были изменены и являются недействительными. Эта операция известна как аннулирование кэш-памяти. Если в кэш-контроллере реализована стратегия отложенной записи и если нужные данные находятся в кэш-памяти, то ее содержимое передается в основную память при особых условиях. Это происходит, например, в том случае, если схема DMA должна передать информацию из основной памяти периферийному устройству, но текущие значения находятся только в кэш-памяти SRAM. Такая операция называется очисткой кэш-памяти.  
  
Программные варианты кэш-памяти, реализованные на компьютере, обладают похожими свойствами. DOS, например, для доступа к гибким и жестким дискам использует внутренние буферы. Можно задать число таких буферов посредством команды BUFFERS в CONFIG.SYS. Эти буферы служат в качестве кэш-памяти, расположенной между процессором, основной памятью и контроллером гибких или жестких дисков. В такой программной кэш-памяти не реализована стратегия сквозной записи, поэтому в случае сбоя системы часть данных, еще не записанных на диск, может находиться в буфере. Только при закрытии обрабатываемого файла или в том случае, если буфер требуется для другого файла или записи, DOS физически записывает содержимое буфера на диск. Выполняя функции 0dh, 5d01h, 68h или выдав прерывание INT 21h, можно заставить DOS выполнить очистку кэш-памяти. С другой стороны, большинство моделей, реализующих программную кэш-памяти для жестких дисков (например, последние версии SMARTDRV, эмулирующие кэш для операций доступа к жесткому диску) в основной памяти, использует стратегию сквозной записи. Все операции записи обращаются при этом к диску, и только чтение производится с копии, размещенной в основной памяти. Новейшие версии SMARTDRV.SYS используют стратегию отложенной записи. В этом случае доступ к диску на запись производится позднее, когда операция отложенной записи может быть выполнена без вмешательства в работу других программ.

**Заключение**

Кэш-память, появившаяся в конце 80-х, первоначально была доступна немногим и зачастую ей пророчили короткое будущее: цены на нее были астрономические, а реальной прибавки быстродействия было недостаточно. Однако бум высоких технологий в 90-х изменил отношение пользователей и специалистов к подобному явлению: сейчас кэш любят и уменьшают зачастую в случае опять же дороговизны со слезами.

Выпустив 486 процессор, фирма Intel произвела настоящую революцию. Здесь не только наличие встроенного математического сопроцессора и умножение частоты, но еще и наличие встроенного кэш. Грустно признать, но с этого момента подобных революций Intel уже не производила.

После этого был Pentium Pro, несущий на себе уже два уровня кэш-памяти, а затем и Pentium II, аналогичный предыдущему, но более быстрый и более дешевый.

Затем наметилось разделение: встроенный кэш второго уровня обеспечивал серьезный подъем производительности, но стоил дорого. Поэтому в дешевых моделях процессоров (Celeron Covington – 266-300 МГц) его устанавливать не спешили (опять же есть мнение, что кэш-то там есть, просто его работа запрещена).

Однако в следующей модели Celeron – Mednocino (300-433 МГц) кэш второй памяти уже был. Процессор Pentium III компании Intel имеет кэш-память первого уровня емкостью 32 Кбайт на микросхеме процессора и либо кэш-память второго уровня емкостью 256 Кбайт на микросхеме, либо кэш-память второго уровня емкостью 512 Кбайт, не интегрированную с процессором.И на сегодняшний день, по крайней мере Intel, больше не заявляет о том, что собирается лишать какой-либо свой будущий процессор этого полезного довеска.

**Список литературы**

* Гук. М. Процессоры intel от 8086 до Pentium 4. С-Питербург  -“Питер Паблишинг” – 2002.
* В.Э. Фигурнов IBM PC для пользователя;  8 издание. Москва - «Финансы», 2001.
* Обзор процессоров и шин ПВМ. Москва – 2001.
* Радовский Н. Пути апгрейда. Компьютерра-Спецвыпуск, Зима 2002, Зимние подарки. С. 14-21.