

**Кэш-память** (от англ. *cache*, дословно — «зачачка», «кубышка», амер.англ. *cash* - «наличные», «деньги под рукою») — память ЭВМ с быстрым доступом, где дублируется часть данных с другого носителя с более медленным доступом, или хранятся данные, для получения которых требуются «дорогие» (в смысле временных затрат) вычисления. Иногда для краткости кэш-память называют просто «кэш».

**Кеш** или **кэш** — это также сленговое обозначение наличных денег.

Собственно, аналогия сравнения кэша (*. cash*) и кеша (*cache*) заключается в следующем: основная память, к которой происходит обращение (жёсткий диск, ОЗУ) сравнивается с вашим счётом в банке, а **кеш** — с вашими наличными, которые вы берёте в банке для быстрых расчётов.

Кэш-память позволяет обращаться к часто требуемым данным быстрее, чем это происходило бы без её использования. Процесс организации доступа через кэш-память называется кэшированием, а та память, которая кэшируется, называется основной памятью.

## Кэширование оперативной памяти

Наиболее часто термин *кеш-память* используется для обозначения [кеш](#)-памяти, находящейся между регистрами центрального процессора (ЦП) и ОЗУ

Кеш-память может давать значительный выигрыш в производительности, потому что в настоящее время тактовая частота ОЗУ значительно меньше тактовой частоты ЦП. Тактовая частота для кеш-памяти обычно не намного меньше частоты ЦП.

## Уровни кеша

Разделение кеш-памяти на несколько уровней (до 3 для универсальных процессоров по состоянию на начало 2009 года). Кеш-память уровня N+1 всегда больше по размеру и медленнее по скорости обращения, чем кеш-память уровня N.

Самой быстрой памятью является кеш-память первого уровня (она же **L1-cache**: **L1I**, **L1D**), по сути, она является **неотъемлемой частью процессора**, поскольку расположена на одном с ним кристалле и входит в состав функциональных блоков, без нее процессор не сможет функционировать. Память L1 работает на частоте процессора и в общем случае обращение к ней может производиться каждый такт (зачастую является возможным выполнять даже несколько чтений/записей одновременно), латентность доступа обычно равна **2-4 такта ядра**, объем этой памяти обычно невелик — **не более 64Кб**.

Второй по быстродействию является **L2** (в отличие от L1 ее можно отключить с сохранением работоспособности процессора), кеш второго уровня, она обычно расположена **либо на кристалле, как и L1, либо в непосредственной близости от ядра**, например, в процессорном картридже (только в слотовых процессорах), в старых процессорах ее располагали на системной плате. Объем L2 по-больше — **от 128Кб до 1—4Мб**. Обычно латентность L2 расположенной на кристалле ядра составляет от **8 до 20 тактов ядра**.

Кеш третьего уровня **L3** наименее быстродействующий и обычно расположен отдельно от ядра ЦП, но он может быть очень внушительного размера и всё равно значительно быстрее, чем оперативная память.

### **Ассоциативность кэша**

Одна из фундаментальных характеристик кэш-памяти - уровень ассоциативности - отображает ее логическую сегментацию. Дело в том, что последовательный перебор всех строк кэша в поисках необходимых данных потребовал бы десятков тактов и свел бы на нет весь выигрыш от использования встроенной в ЦП памяти. Поэтому ячейки ОЗУ жестко 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 (*n*-way set associative) обозначает, что информация по некоторому адресу оперативной памяти может храниться в *n* мест кэш-памяти.

При одинаковом объеме кэша схема с большей ассоциативностью будет наименее быстрой, но наиболее эффективной.

### **Политика записи при кешировании**

При чтении данных кеш-память даёт однозначный выигрыш в производительности. При записи данных выигрыш можно получить только ценой снижения надёжности. Поэтому в различных приложениях может быть выбрана та или иная политика записи кеш-памяти..

Существуют две основные политики записи кеш-памяти — сквозная запись (*write-through*) и отложенная запись (*write-back*).

- сквозная запись подразумевает, что при изменении содержимого ячейки памяти, запись происходит синхронно и в кеш и в основную память.
- отложенная запись подразумевает, что можно отложить момент записи данных в основную память, а записать их только в кеш. При этом данные будут выгружены в оперативную память только в случае обращения к ним какого либо другого устройства (другой ЦП контроллер DMA) либо нехватки места в кеше для размещения других данных. Производительность, по сравнению со сквозной записью, повышается, но это может поставить под угрозу целостность данных в основной памяти, поскольку программный или аппаратный сбой может привести к тому, что данные так и не будут переписаны из кеша в основную память. Кроме того, в случае кеширования оперативной памяти, когда используются два и более процессоров, нужно обеспечивать согласованность данных в разных кешах.

### **Кеширование внешних накопителей**

Многие периферийные устройства хранения данных используют кеш для ускорения работы, в частности, жёсткие диски используют кеш-память от 1 до 16 Мб (модели с поддержкой NCQ/TCQ используют её для хранения и обработки запросов), устройства чтения CD/DVD/BD-дисков так же кешируют прочитанную информацию для ускорения повторного обращения.

Операционная система так же использует часть оперативной памяти в качестве кеша дисковых операций (в том числе для внешних устройств, не обладающих собственной кэш-памятью, например, USB-flash, дисковод для дискет).

### **Кеширование интернет-страниц**

Процесс сохранения часто запрашиваемых документов на промежуточных прокси-серверах или машине пользователя, с целью предотвращения их постоянной загрузки с сервера-источника и уменьшения трафика. Т.е. перемещение информации поближе к пользователю. Управление кэшированием осуществляется при помощи HTTP-заголовков

### **Кеширование результатов работы**

Многие программы записывают куда-либо промежуточные или вспомогательные результаты работы, чтобы не вычислять их каждый раз, когда они понадобятся. Это ускоряет работу, но требует дополнительной памяти (оперативной или дисковой). Примером такого кеширования является индексирование баз данных.

## **Уровень за уровнем**

Хотя оперативная память намного быстрее диска, тем не менее и она не успевает за потребностями процессора. Поэтому данные, которые требуются часто, переносятся на следующий уровень быстрой памяти, называемой кэш-памятью второго уровня. Она может располагаться на отдельной высокоскоростной микросхеме статической памяти (SRAM), установленной в непосредственной близости от процессора (в новых процессорах кэш-память второго уровня интегрирована непосредственно в микросхему процессора).

На более высоком уровне информация, используемая чаще всего (скажем, команды в многократно выполняемом цикле), хранится в специальной секции процессора, называемой кэш-памятью первого уровня. Это самая быстрая память.

Процессор Pentium III компании Intel имел 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.

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.

Когда процессор обнаруживает данные в одном из кэшей, это называют «попаданием»; неудачу называют «промахом». Каждый промах вызывает задержку, поскольку процессор будет пытаться обнаружить данные на другом, более медленном уровне. В хорошо спроектированных системах с программными алгоритмами, которые выполняют предварительную выборку данных до того, как они потребуются, процент «попаданий» может достигать 90.

Для процессоров старшего класса на получение информации из кэш-памяти первого уровня может уйти от одного до трех тактов, а процессор в это время ждет и ничего полезного не делает. Скорость доступа к данным из кэш-памяти второго уровня, размещаемой на процессорной плате, составляет от 6 до 12 циклов, а в случае с внешней кэш-памятью второго уровня — десятки или даже сотни циклов.

Кэш-память для серверов даже более важна, чем для настольных ПК, поскольку серверы поддерживают между процессором и памятью весьма высокий уровень трафика, генерируемого клиентскими транзакциями. В 1991 году Intel превратила ПК на базе процессора 80486 с тактовой частотой 50 МГц в сервер, добавив на процессорную плату кэш с тактовой частотой 50 МГц. Хотя шина, связывающая процессор и память, работала с частотой всего 25 МГц, такая кэш-память позволила многие программы во время работы полностью размещать в процессоре 486 с тактовой частотой 50 МГц.

Иерархическая организация памяти помогает компенсировать разрыв между скоростями процессоров, ежегодно увеличивающимися примерно на 50% в год, и скоростями доступа к DRAM, которые растут лишь на 5%.

По мере усиления этого диссонанса производители аппаратного обеспечения добавляют третий, а возможно и четвертый уровень кэш-памяти.

В 2000 году Intel представила кэш-память третьего уровня в своих 64-разрядных процессорах Itanium. Кэш емкостью 2 или 4 Мбайт будет связан с процессором специальной шиной, тактовая частота которой совпадает с частотой процессора.

IBM также разработала собственную кэш-память третьего уровня для 32- и 64-разрядных ПК-серверов Netfinity.

Сначала кэш будет размещаться на микросхеме контроллера памяти.

Кэш-память третьего уровня корпорации IBM стала общесистемным кэшем, куда могут обращаться от 4 до 16 процессоров сервера. С кэш-памятью третьего уровня Intel может работать только тот процессор, к которому она подключена, но представители IBM подчеркнули, что их кэш третьего уровня способен увеличить пропускную способность всей системы. Новая кэш-память производства IBM также поможет реализовать компьютерные системы высокой готовности, необходимые для электронной коммерции, поскольку с ее помощью можно будет менять модули основной памяти и выполнять модернизацию, не прерывая работу системы.

## **Больше – не всегда лучше**

Частота промахов при обращении к кэш-памяти может быть значительно снижена за счет увеличения емкости кэша. Но большая кэш-память требует больше энергии, генерирует больше тепла и увеличивает число бракованных микросхем при производстве.

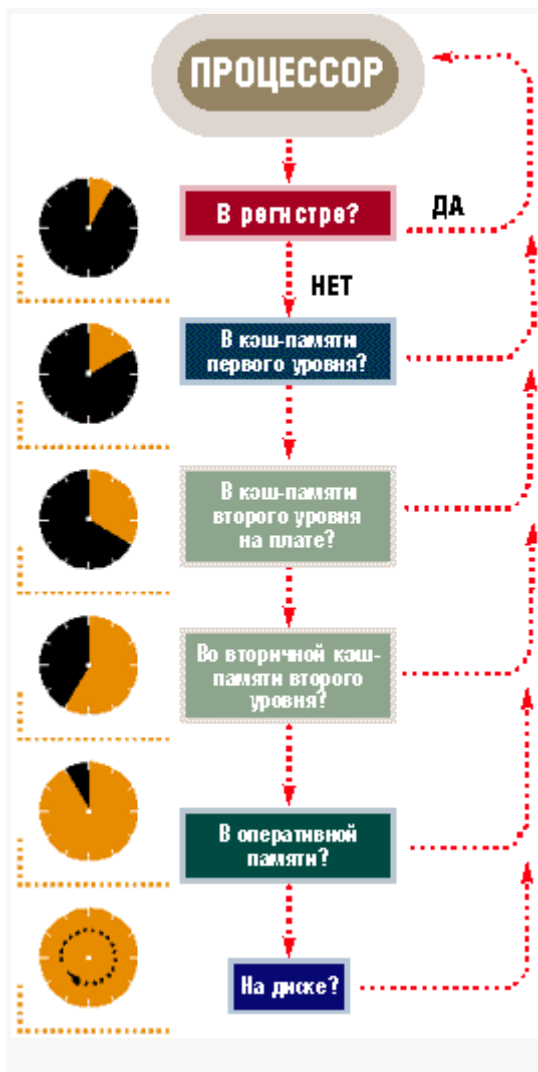
Один из способов обойти эти трудности — передача логики управления кэш-памятью от аппаратного обеспечения к программному.

«Компилятор потенциально в состоянии анализировать поведение программы и генерировать команды по переносу данных между уровнями памяти»

Управляемая программным образом кэш-память сейчас существует лишь в исследовательских лабораториях. Возможные трудности связаны с тем, что придется переписывать компиляторы и перекомпилировать унаследованный код для всех процессоров нового поколения.

---

## **Где мои данные?**



Когда процессору требуются данные, он сначала анализирует содержимое своих регистров данных. Если данных там нет, процессор смотрит, не лежат ли они в ближайшей к нему кэш-памяти первого уровня. Если и там нет, то следующее обращение происходит к кэш-памяти второго уровня. Если процессор не находит данных в кэше, он проверяет оперативную память. И здесь нет? Тогда процессор посылает запрос к диску. Время идет, а процессор ничего полезного не делает...