# Кеш памет

## I. Въведение

Кеш (cache) паметта е тип статична RAM (SRAM) която се използва с цел подобряване на производителността на системи при които се налага комуникация между две устройства, от които едното е в пъти по-бързо от другото. Подобни системи са неефективни от гледна точка на бързодействие, тъй като „бързото" устройство трябва да изчаква „бавното" устройство всеки път, когото очаква да получи данни от него. В този случай бързодействието на системата клони към бързодействието на „бавното" устройство. Едно възможно решение е използването на кеш памет като буферна памет, която физически се намира между „бързото" и „бавното" устройства. Целта е „бързото" устройство да комуникира само с бързата кеш памет. В идеалния случай, бързодействието на кеш паметта и на „бързото" устройство трябва да съвпадат. Така не се налага изчакване и обменът е с максимална скорост.

Когато разликата в бързодействието между „бързото" и „бавното" устройство е много голяма се налага да се използват няколко нива на кеширане на информацията. Кеш паметта, която е най-близо до „бързото" устройство, се нарича кеш от ниво 1 (L1 кеш). Кеш паметта, която е най-близо до „бавното" устройство, се нарича кеш от ниво *n* (Ln кеш). При персоналните компютри най-често се използва кеш с 2 до 4 нива на кеширане на информацията. Тъй като кеш паметта е скъпа, в сила са следните ограничения за различните нива на кеш: най-бърза и с най-малък размер е L1 кеш, а Ln кеш е с най-голям размер, но и най-бавна.

При персоналните компютри кеш паметта се използва основно с цел буфериране на обмена между микропроцесора (бързо устройство) и оперативната памет (бавно устройство). Кеш памет се използва и при всички периферни устройства при които има съществена разлика в бързодействието на модулите в тях, които си комуникират. Например, при твърдите дискове се използва кеш памет, тъй като има съществена разлика в бързодействието на интерфейса, чрез който диска се свързва към дънната платка (хост адаптер), и механиката на диска (четящо-записващи глави).

При микропроцесорите L1 кеш винаги е вградена в чипа с цел максимално бързодействие между CPU и L1 кеш. По този начин се намалява дължината на пътечките, свързващи двата модула и това води до намаляване на електромагнитните смущения. Има микропроцесори при които в чипа са вградени две нива на кеш (L1 и L2). Има и микропроцесори с три нива на кеш (L1, L2 и L3). Микропроцесорът има физическа връзка само с L1 кеш и извлича инструкции и данни от него. Следователно, друго устройство трябва да зарежда L1 кеш с необходимата информация. Това е модул от чипсета, по точно контролера за управление на паметта, част от северния мост (при мостова архитектура на чипсета) или Memory Controller Hub (при хъбова архитектура на чипсета). Този модул се грижи за синхронизиране на обмена между CPU-Cache-DRAM. Обменът на информация между CPU и чипсета се реализира по специална високоскоростна паралелна шина – Front Side Bus (FSB), а обменът между чипсета и DRAM – чрез друга специална паралелна шина, която най-често е 64-битова.

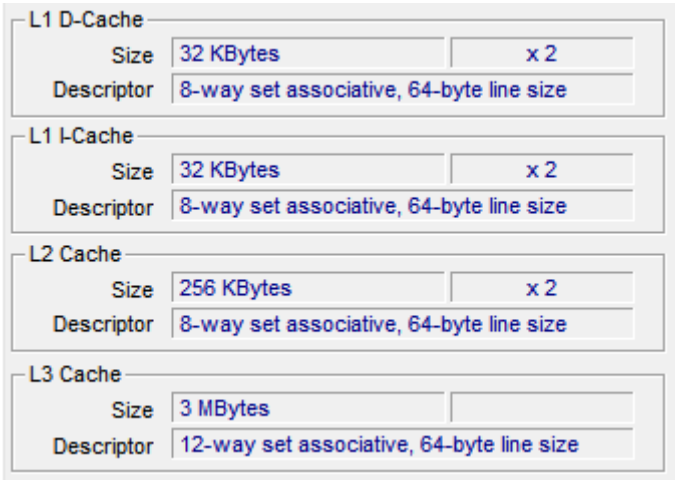
## II. Архитектурни решения

Всеки микропроцесор изпълнява инструкции, а те оперират с данни. Инструкциите и данните формират програма. Когато една програма се зареди от операционната система в DRAM с цел изпълнение тя става процес. При съвременните многозадачни операционни системи микропроцесорът работи със всеки процес за определен интервал от време. С помощта на чипсета, инструкциите и данните от DRAM достигат до L1 кеш. Използват се две основни архитектурни решения за L1 кеш:

- Кеш с архитектура Принстън;
- Кеш с архитектура Харвард.

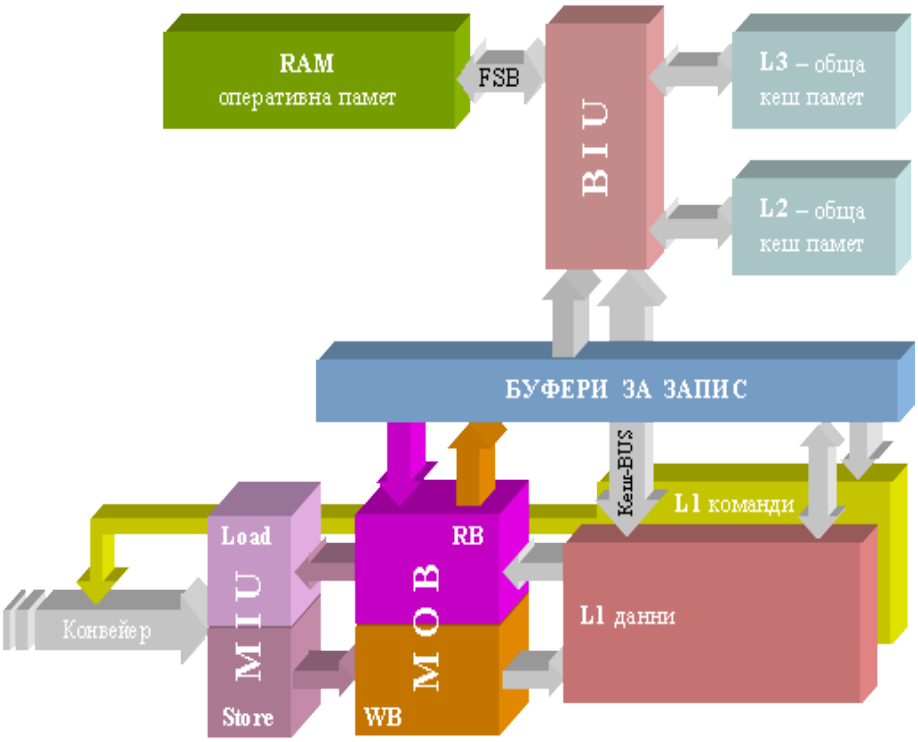
При архитектура Принстън кеш паметта е обща – кеш за инструкции и за данни. При архитектура Харвард кеш паметта се разделя физически на кеш само за инструкции (instruction cache) и кеш само за данни (data cache). От гледна точка на бързодействие по-ефективна е архитектура Харвард при един и същ капацитет на паметта. Това е така, тъй като кеша за инструкции и кеша за данни се достъпват паралелно (до всеки кеш модул има отделни шини за данни и адреси). С цел допълнително подобряване на бързодействието е възможно кеш паметта за данни да е двупортова. Така става възможно за един такт да се извлече операнд от кеша за инструкции и един или два операнда от кеша за данни. По-голямата част от микропроцесорите за персонални компютри имат еднопортова кеш памет с цел понижаване на цената. Кеш паметта от горните нива (L2, L3) е обща кеш за инструкции и данни.

На Фиг. 1 е показано описание на кеш паметта на компютър с микропроцесор Intel Core i3. Системата има три нива на кеш. Кешът L1 е с архитектура Харвард (L1 D-Cache и L1 I-Cache), а L2 и L3 кеш са с архитектура Принстън – общ кеш за инструкции и данни. Ясно се вижда, че размерът на кеш паметта от различните нива е обратно пропорционален на неговата цена (L1 = 2 x32 KiB, L2=256 KiB и L3=3 MiB). За всяко ниво на кеш, без последното (L3), има по два модула (x 2). Това е характерно архитектурно решение за многоядрените микропроцесорите на Intel – всяко ядро има собствена кеш памет (в случая L1 и L2 кеш), а последното ниво кеш (в случая L3 кеш) е общо за всички ядра. Достъпът до кеш паметта е на ниво кеш ред (line size). Един кеш ред се формира от *n* на брой последователни байта от DRAM. При конкретния пример (виж Фиг. 1) размерът на един кеш ред е 64 байта. Следователно, обменът между CPU-Cache-DRAM е блоков. Размерът на блока съвпада с размера на кеш реда.




**Фиг. 1**. *Кеш памет на компютър с микропроцесор Intel Core i3*

На Фиг. 2 е показана структурата на система с три нива на кеш. Примерът е информативен, а не за реален микропроцесор. Конфигурацията описва кеш с архитектура, показана на Фиг. 1 – три нива на кеш, от които L1 е с архитектура Харвард, а останалите - с архитектура Принстън.




**Фиг. 2**. *Структура на система с три нива на кеш с буфериране при микропроцесор със скаларна (конвейерна) архитектура*

Основните модули, които реализират комуникацията DRAM-Cache-CPU, са следните:

- Memory Interfaces Unit (MIU) – модул за комуникация между конвейера на микропроцесора и кеш паметта.
- Memory Order Buffer (MOB) – буфер за подреждане (при четене и запис).
- Bus Interfaces Unit (BIU) - блок за връзка с процесорната шина (FSB).

Конвейерът изпълнява потока от инструкции, които получава от L1 кеш паметта за инструкции (команди). Модулът MIU е Load-Store памет, която изпълнява функцията на интерфейс между конвейерите и кеш паметта. Операция четене (Load) позволява бързо извличане на желаната информация от L1 кеш или от буферите за запис. Операция запис (Store) е свързана с получаване на резултат от конвейера на микропроцесора. Посредством MIU, резултатът, получен след изпълнение на инструкциите, достига до MOB. Модул MOB е буфер за данни от тип опашка (FIFO). Целта на този буфер е да предотврати забавяне на системата с кеш памет, поради крайното време необходимо за синхронизиране на операции четене / запис от / в кеш паметта. Размерът на тези буфери е ограничен. При запълване на буферите, конвейерите временно трябва да се блокират. За да се ограничи загубата на време, вследствие на блокировките, се налага изчистване на MOB буферите когато е възможно, дори да не са запълнени изцяло. Когато резултатът от изпълнение на инструкциите не е необходим на други програмни модули (нишки), той се записва в буфери за запис, а не в L1 кеш. Тези буфери са необходими, за да се намалят циклите на запис в кеш паметта. Организацията на буферите е такава, че те могат да

помнят *n* на брой кеш реда (от няколко до десетки). В даден момент от време обаче е необходимо съдържанието им да бъде прехвърлено в кеш, например:

- Всички буфери за запис са запълнени, а е необходимо да се запише нов резултат;
- В буфера има резултат, който все още не е в L1 кеш, но е необходим за друг програмен модул (нишка). В този случай се налага принудително обновяване на кеш, за да могат програмните модули да работят с валидни резултати.
- Програмно инвалидиране на кеш паметта чрез специална инструкция (INVD, WBINVD и INVLPG).

Модул BIU реализира интерфейса между процесорната шина (FSB) и кеш паметта. Модулът може да чете / записва от / в кеша от горните нива (L2 и L3), както и буферите за запис.

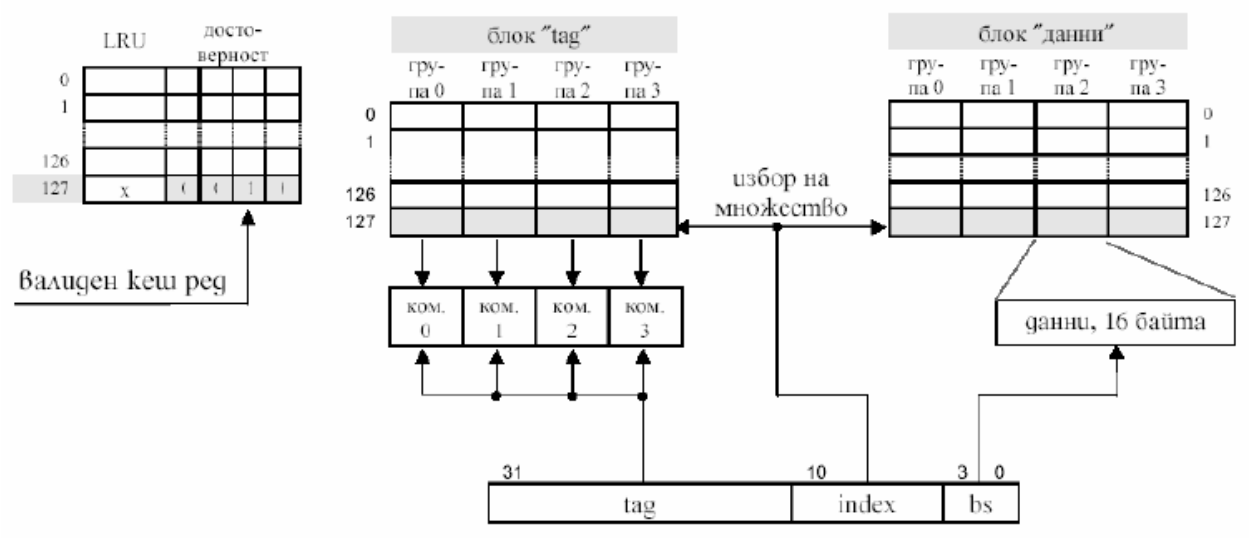
**III. Асоциативна и неасоциативна кеш памет**

Микропроцесорите не адресират директно физическата памет, тъй като нямат регистри с необходимата разрядност. Поради тази причина се използват методи за индиректно адресиране на физическата памет, например сегментиране и странициране.

Съвременните операционни системи реализират достъп до паметта на ниво страници. Размерът на страниците зависи от микропроцесора и на настоящия етап може да бъде 4 KiB или 4 MiB. Всеки процес заема няколко блока от оперативната памет. Всеки блок се състои от една или повече страници. Тези блокове се заделят и освобождават от операционната система. В блоковете може да има: инструкции, статични данни, динамични данни и стек. Системата за управление на кеш паметта трябва да буферира в кеш малки части от всеки блок, в зависимост от това какво е необходимо на микропроцесора в даден момент от време. В зависимост от това от колко блока едновременно може да се кешира информация, кеш паметта се дели на неасоциативна (с директна адресация) и асоциативна. Неасоциативната кеш е с най-проста архитектура и съответно най-евтина. Тя обаче не може да гарантира добра производителност. Причината за това е, че се налага често презареждане на кеш паметта при необходимост за зареждане в кеш на данни или инструкции от различни физически адреси. Това обаче често се налага, тъй като на практика програмния код не е линеен, а разклонен – има условни и безусловни преходи, както и викане на функции (методи). Аналогичен е проблема и с данните – често се налага едновременно да се работи с множество масиви от данни, които физически са разположени е различни блокове от DRAM.

Кеш паметта, която позволява да се кешира информация от множество физически адреси едновременно, се нарича асоциативна. Асоциативната кеш се дели на напълно асоциативна и асоциативна в няколко направления кеш. Най-висока ефективност има напълно асоциативната кеш, тъй като при нея броят на физическите адреси от които може да се кешира зависи само от размера на кеш паметта (размера на кеш / размера на кеш реда). На практика напълно асоциативната кеш се използва по-рядко, поради нейната висока цена. При микропроцесорите за персонални компютри се използва кеш, която е асоциативна в *n* направления (*n*-way set associative cache). При тази кеш памет е възможно кеширане от *n* физически адреса едновременно и така се прави компромис между бързодействие и цена.

Първият микропроцесор с интегрирана L1 кеш е Intel 80486DX. Тя е обща кеш за инструкции и данни с размер 8 KiB. Паметта е асоциативна в 4 направления, а размерът на един кеш ред е 16 байта. На Фиг. 3 е показана вътрешната архитектура на кеш паметта на този микропроцесор.




**Фиг. 3**. *Архитектура на кеш паметта на микропроцесор Intel 80486DX*

Информацията, която се записва е кеш, се организира в матрица, наречена блок „данни". Редовете на матрицата се наричат множества, а стълбовете – групи. Броят на групите е равен на броя на направленията в които е асоциативна паметта. Броят на множествата зависи от общия размер на кеш паметта, броя на направленията и размера на кеш реда:

$$8*1024 / 4 / 16 = 128.$$

За да се определи дали един кеш ред е в блок „данни" или не е се използва блок „tag" и адресни компаратори. Броят на компараторите съвпада с броя на групите. Всяка клетка на блок „tag" съдържа старшите 21 бита на адреса, който вече е бил кеширан.

Микропроцесорът адресира необходимата информация чрез генериране на линеен адрес. В случая той е 32-битов и се дели на три части:

- Поле byte select (bs) – избор на байт от адресирания кеш ред;
- Поле index – избор на множество;
- Поле tag – старшите 21 бита от линейния адрес.

Ако стойността на битовете от поле „tag" на линейния адрес и стойността на адреса кеширан в някоя от групите на избрано множество от блок „tag" съвпадне, то информацията е кеширана. Ако нито един от компараторите не сработи – информацията не е кеширана и трябва да се извлече от DRAM.

Проблемна е ситуацията, когато микропроцесора адресира информация, която няма как да се запише веднага в кеш, тъй като всички групи от избрано множество са заети. В този случай се налага да се освободят данните от дадена група (записват се в DRAM), за да могат да се кешират новите данни. За да се минимизира времето за освобождаване на кеш ред се използват различни стратегии, например:

- Least Recently Used (LRU) – при тази стратегия се освобождава се реда, който най-дълго не е бил адресиран от микропроцесора.

- Least Recently Allocated (LRA) – освобождава се блокът, който най-отдавна е бил кеширан за първи път.

За да може да функционира стратегията за освобождаване на кеш ред, при микропроцесор Intel 80486DX се използват битовете от блок LRU (един запис за всяко множество).

### IV. Съгласуваност на паметта

Ефективността на кеш паметта се измерва с параметър наречен коефициент на попадение в кеш. При заявка от страна на процесора за определена информация имаме кеш попадение (hit), ако тази информация е вече кеширана. В противен случай се казва, че има кеш пропуск (miss). Ясно е, че колкото повече последователни попадения има в кеш паметта, толкова по-ефективна е системата с кеш памет. Коефициентът на попадение зависи от множество фактори, например:

- Размер на кеш паметта;
- Асоциативност на кеш паметта;
- Многоядрен или едноядрен е микропроцесора;
- Стратегия за освобождаване на кеш ред и много др.

Основният проблем при системите с кеш памет е гарантиране на съгласуваност (кохерентност) на паметта. Паметта е съгласувана, когато няма разлика между съдържанието на клетките от DRAM и съответните им копия е кеш паметта. Ако трябва да се гарантира постоянна съгласуваност на паметта се налага постоянно обновяване на всички нива на кеш и DRAM при всеки запис. Това е неефективно, тъй като ще срине производителността на компютърната система. Съвременните компютърни системи гарантират съгласуваност на паметта, но само в моментите когато тя е необходима (отложена съгласуваност), например при промяна на стойността на споделени данни. Споделени са данните, които се използват от повече от един програмен модул едновременно. Типичен пример за това е достъп до споделени данни от няколко програмни нишки едновременно.

Съществуват множество патентовани методи, които се използват от фирмите производители на микропроцесори, които гарантират отложена съгласуваност на паметта. За целта се използват специални методи за кеширане и протоколи за кеширане на информацията при запис. Четенето е безпроблемна операция, която се реализира лесно. Проблемът със съгласуваността на паметта е още по-голям при многоядрените микропроцесори. При тях споделените данни могат да имат различни стойности в различните нива на кеш в различните ядра.

#### 4.1. Протокол MESI (Modified, Exclusive, Shared, Invalid)

Един от най-често използваните протоколи чрез които се задават състояния за кеш редовете с цел гарантиране на съгласуваност на кеш за данни е MESI. От името на протокола следва, че всеки кеш ред може да бъде в едно от 4 възможни състояния:

- Modified (променено) - съдържанието на кеш-линията е обновено, но промяната не е реализирана в DRAM. Редът няма копие в кеша на други процесори (ядра). Записът в реда не води до запис в DRAM.

- Exclusive (изключително) – образът на кеш реда в DRAM съдържа валидна информация. Редът няма копие в други процесори (ядра). Записът в реда не води до запис в DRAM, но се състоянието му се променя до Modified.
- Shared (споделено) - образът на реда в DRAM съдържа валидна информация. Възможно е обаче редът да има копие в кеш на други процесори (ядра).
- Invalid (невалидно) - данните в този кеш ред са невалидни (няма данни).

За да се гарантира съгласуваност на кеш паметта за инструкции не са необходими всички 4 състояния, а само две - Shared и Invalid. Останалите състояния не са необходими, тъй като програмния код се записва в Read Only страници (забранено е модифициране или само-модифициране на програмния код).

В Табл. 1 е показана спецификата на данните при всяко едно от състоянията, които протокол MESI дефинира.

**Табл. 1.** *Специфика на данните за всяко състояние на кеш редовете при протокол MESI*

| **Състояние** | **Modified** | **Exclusive** | **Shared** | **Invalid** |
|---|---|---|---|---|
| Валиден ли е реда? | да | да | да | не |
| Образът на кеш реда в DRAM валиден ли е? | не | да | да | Няма образ в DRAM |
| Кеш реда има ли копие в кеш на други процесори (ядра)? | не | не | Възможно е | Възможно е |
| Какво става при запис в кеш реда? | Обновява се кеш реда, но не и DRAM | Обновява се кеш реда | Обновява се кеш реда, възможно е обновяване на копието му в други кеш памети | Запис в кеш реда |

### 4.2. Методи за кеширане

Методите за кеширане дефинират какво се случва при всеки запис в кеш паметта. Няма стандартни методи за кеширане - всяка фирма разработва свои. След като се стартира, операционната система може да зададе различен метод за кеширане за различните блокове от паметта. Изборът на метод зависи от това каква точно информация се очаква да има във всеки един от блоковете (дали се променя или не, колко често се очаква да се променя, данни или инструкции ще има в блока и др.). Най-често използваните методи за кеширане са следните:

- Wright Through (WT) – При този метод се гарантира съгласуваност на паметта по всяко време, тъй като след всеки запис в L1 кеш ред следва запис в горните нива на кеш и DRAM. Методът е неефективен при много на брой последователни записи. За да се минимизират циклите на обмен с цел валидиране на кеш и оперативната памет на практика се използва метод WT с буфериране. При този метод, всеки запис в кеш води само до запис в специален буфер. След запълването му се реализира цикъл на обновяване на паметта от горните нива на кеш до DRAM.
- Write Back (WB) – При този метод на кеширане се гарантира отложена съгласуваност на паметта. Записът не информация в L1 кеш ред не води веднага до

обновяване на горните нива кеш и DRAM. Обновяването се реализира само, ако е наложително, например съдържанието на кеш реда в който се записва има копие в кеша на друг процесор. На практика се използва метод WB с буфериране.

- Uncachable (UC) – Този метод се задава за блокове от паметта, които съдържат данни, които не трябва да се кешират. Най-често това са блокове с множество споделени данни, които не е подходящо да се кешират защото често се модифицират.

Всяка фирма, производител на микропроцесори, използва много други методи за кеширане. Те могат да се различават по начина на функциониране при различните семейства микропроцесори на съответния производител.

**V. Въпроси и задачи за самостоятелна работа**

1. Къде, освен при CPU, се използва кеш памет при персоналните компютри?

2. От какво зависи колко нива на кеш ще се използват при една компютърна система?

3. Къде физически може да се намират отделните нива на кеш?

4. Използвайте фирмена литература и проверете как се гарантира съгласуваност на паметта при микропроцесори от семейство Intel Core ix.

5. Какви методи за кеширане използват фирми Intel и AMD при своите микропроцесори?

6. Използвайте фирмена литература и проверете как точно функционира стратегия LRU.

7. Колко адресни компаратора има кеш памет с капацитет 32 KiB при размер на кеш реда от 64 байта, ако паметта е неасоциативна, асоциативна в 8 направления и напълно асоциативна?