

Загальні принципи функціонування кеш-пам'яті

Кеш-пам'ять (КП), або кеш, являє собою організовану у вигляді асоціативної пам'яті (АЗП) швидкодіючу буферну пам'ять обмеженого обсягу, яка розташовується між регістрами процесора і відносно повільною основною пам'яттю і зберігає найбільш часто використовувану інформацію спільно з її ознаками (тегами), в якості яких виступає частина адресного коду.

В процесі роботи окремі блоки інформації копіюються з основної пам'яті в кеш-пам'ять. При звертанні процесора за командою або даними спочатку перевіряється їх наявність в КП. Якщо необхідна інформація знаходиться в кеші, вона швидко витягується. Це кеш-попадання. Якщо необхідна інформація в КП відсутня (кеш-промах), то вона вибирається з основної пам'яті, передається в мікропроцесор і одночасно заноситься в кеш-пам'ять. Підвищення швидкодії обчислювальної системи досягається в тому випадку, коли кеш-попадання реалізуються набагато частіше, ніж кеш-промахи.

Механізм збереження інформації в кеш-пам'яті

При включенні мікропроцесора в роботу вся інформація в його кеш-пам'яті недостовірна. При зверненні до пам'яті мікропроцесор, як уже зазначалося, спочатку перевіряє, чи не міститься шукана інформація в кеш-пам'яті. Для цього сформована ним фізична адреса порівнюється з адресами осередків пам'яті, які були раніше кешованими з ОЗП в КП.

При першому зверненні такої інформації в кеш-пам'яті, природно, немає, і це відповідає кеш-промаху. Тоді мікропроцесор проводить звернення до оперативної пам'яті, витягує потрібну інформацію, використовує її у своїй роботі, але одночасно записує цю інформацію в кеш.

Якби в кеш-пам'ять заносили тільки інформацію, яку вимагає мікропроцесор в даний момент, то, швидше за все, при наступному зверненні знову стався б кеш-промах: навряд чи наступне звернення станеться до тієї ж самої команди або до того ж самому операнду. Кеш-попадання відбувалися б лише після того, як в КП накопичиться достатньо великий фрагмент програми, що містить деякі циклічні ділянки коду, або фрагмент даних, які підлягають повторній обробці. Для того щоб

вже наступне звернення до КП призводить якомога частіше до кеш-влучень, передача з оперативної пам'яті в кеш-пам'ять відбувається не тими порціями (байтами або словами), які затребувані мікропроцесором в даному зверненні, а так званими рядками. Тобто кеш-пам'ять і оперативна пам'ять з точки зору кешування організовуються у вигляді рядків. Довжина рядка перевищує максимально можливу довжину затребуваних мікропроцесором даних. Зазвичай вона становить від 16 до 64 байт і вирівняна в пам'яті по межі відповідного розділу (рис. 4.1).

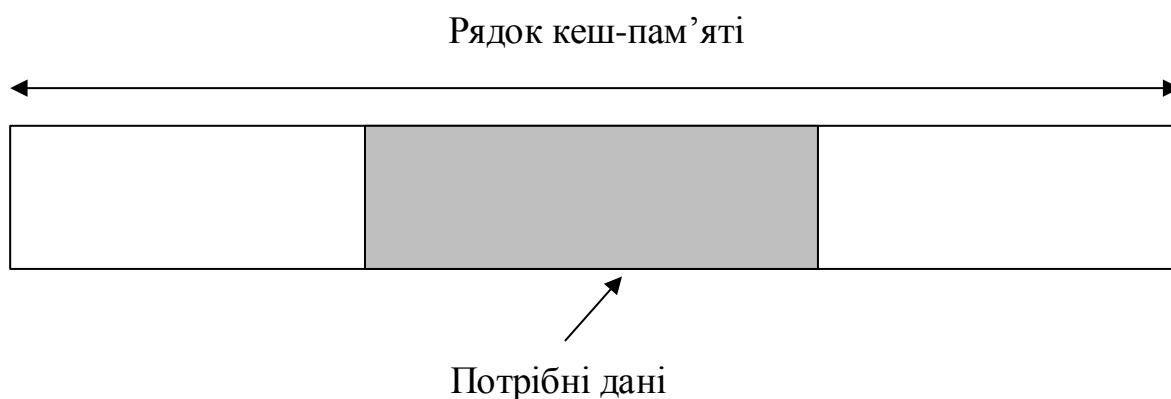


Рис. 4.1. Організація обміну між оперативної і кеш-пам'яттю

Високий відсоток кеш-влучень в цьому випадку забезпечується завдяки тому, що в більшості випадків програми звертаються до осередків пам'яті, розташованим поблизу від раніше використаних. Ця властивість, називається принципом локальності посилань, вона забезпечує ефективність використання КП. Мається на увазі, що при виконанні програми протягом деякого малого інтервалу часу відбувається звернення до пам'яті в межах обмеженого діапазону адрес (як за кодом програми, так і за даними).

Наприклад, мікропроцесору для своєї роботи знадобилося 2 байти інформації. Якщо рядок має довжину 16 байт, то в кеш переписуються не тільки потрібні 2 байти, а й деяке їх оточення. Коли мікропроцесор звертається за новою інформацією, в силу локальності посилань, швидше за все, звернення станеться за сусідньою адресою. Потім знову за сусідньою, знову за сусідньою і т. д.. Таким чином, ряд наступних звернень буде відбуватися безпосередньо до кеш-пам'яті, минаючи оперативну пам'ять (кеш-попадання). Коли чергова сформована мікропроцесором фізична адреса

вийде за межі рядка кеш-пам'яті (відбудеться кеш-промах), буде виконано підкачка в кеш нового рядка, і знову ряд наступних звернень викличе кеш-попадання.

Чим довший використовується при обміні між оперативною та кеш-пам'яттю рядок, тим більша ймовірність того, що наступне звернення станеться в межах цього рядка. Але в той же час чим довший рядок, тим довше він буде перекачуватися з оперативної пам'яті в кеш. І якщо чергова команда виявиться командою переходу або вибірка даних почнеться з нового масиву, тобто наступне звернення відбудеться не за сусідньою адресою, то час, витрачений на передачу довжини рядка, буде використано марно. Тому при виборі довжини рядка повинен бути розумний компроміс між співвідношенням часу звернення до оперативної і кеш-пам'яті і ймовірністю досить віддаленого переходу від поточної адреси при виконанні програми. Зазвичай довжина рядка визначається в результаті моделювання апаратно-програмної структури системи.

Після того як в КП накопичиться достатньо великий обсяг інформації, збільшується ймовірність того, що формування чергової адреси призведе до кеш-попадання. Особливо велика ймовірність цього при виконанні циклічних ділянок програми.

Стара інформація по можливості зберігається в кеш-пам'яті. Її заміна на нову визначається ємністю, організацією і стратегією оновлення кешу.

Типи кеш-пам'яті

Якщо кожен рядок ОЗП має тільки одне фіксоване місце, на якому він може перебувати в кеш-пам'яті, то така кеш-пам'ять називається пам'яттю з **прямим відображенням**.

Припустимо, що ОЗП складається з 1000 рядків з номерами від 0 до 999, а кеш-пам'ять має ємність тільки 100 рядків. У кеш-пам'яті з прямим відображенням рядки ОЗП з номерами 0, 100, 200, ..., 900 можуть зберігатися тільки в рядку 0 КП і ніде інакше, рядки 1, 101, 201, ..., 901 - в рядку 1 КП (рис. 4.2). Така організація кеш-пам'яті забезпечує швидкий пошук в ній потрібної інформації: необхідно перевірити її наявність тільки в одному місці. Однак ємність КП при цьому використовується не в повній мірі: не дивлячись на те, що частина кеш-пам'яті може бути не заповнена, буде

відбуватися витіснення з неї корисної інформації при послідовних зверненнях, наприклад, до рядків 101, 301, 101 ОЗП.

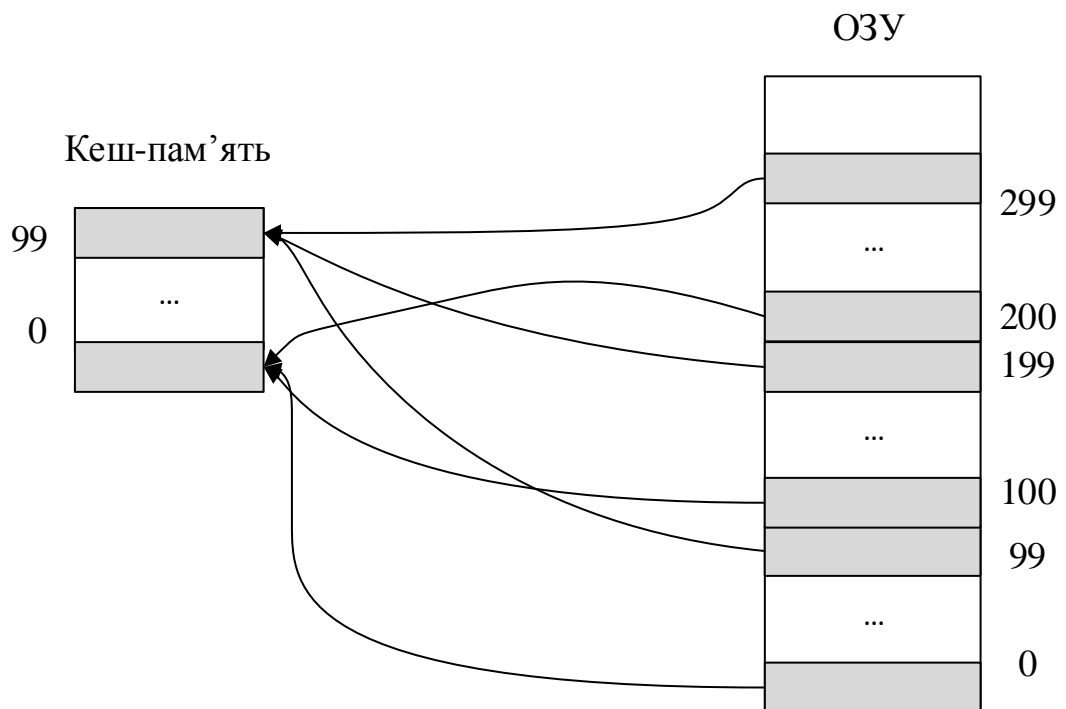


Рис. 4.2. Принцип організації кеш-пам'яті з прямим відображенням

Кеш-пам'ять називається повністю асоціативною, якщо кожен рядок ОЗП може розташовуватися в будь-якому місці кеш-пам'яті.

У повністю асоціативній кеш-пам'яті максимально використовується весь її об'єм: витіснення збереженої в КП інформації проводиться лише після її повного заповнення. Однак пошук в кеш-пам'яті, організованої подібним чином, є важким завданням.

Компромісом між цими двома способами організації кеш-пам'яті є множинно-асоціативна КП, в якій кожен рядок ОЗП може перебувати по обмеженій кількості місць в кеш-пам'яті.

При необхідності заміщення інформації в кеш-пам'яті на нову використовується кілька стратегій заміщення. Найбільш відомими серед них є:

- LRU - заміщається рядок, до якого найдовше не було звернень;
- FIFO - заміщається найдавніший по перебуванню в кеш-пам'яті рядок;
- Random - заміщення проходить випадковим чином.

Останній варіант, суттєво заощаджує апаратні засоби в порівнянні з іншими підходами, в ряді випадків забезпечує і більш ефективне використання кеш-пам'яті. Припустимо, наприклад, що КП має об'єм 4 рядки, а деяка циклічна ділянка програми має довжину 5 рядків. В цьому випадку при стратегіях LRU і FIFO кеш-пам'ять виявиться фактично марною через відсутність кеш-влучень. У той же час при використанні стратегії випадкового заміщення інформації частина звернень до КП призведе до кеш-влучень.

Деякі евристичні оцінки ймовірності кеш-промаху при різних стратегіях заміщення (у відсотках) представлені в табл. 4.1.

Таблиця 4.1. Ймовірність кеш-промаху для різної кеш-пам'яті

Організація кеш-пам'яті

Розмір кешу, Кбайт	2-канальна асоціативна		4-канальна асоціативна		8-канальна асоціативна	
	<i>LRU</i>	Random	<i>LRU</i>	Random	<i>LRU</i>	Random
16	5.2	5.7	4.7	5.3	4.4	5.0
64	1.9	2.0	1.5	1.7	1.4	1.5
256	1.15	1.17	1.13	1.13	1.12	1.12

Аналіз таблиці показує, що:

- зі збільшенням ємності кешу, природно, зменшується ймовірність кеш-промаху, і на сьогоднішній день близько 95% звернень відбуваються до КП, минаючи оперативну пам'ять;
- чим більший ступінь асоціативності кеш-пам'яті, тим більша ймовірність кеш-попадання за рахунок більш повного заповнення КП (час пошуку інформації в КП в даному аналізі не враховується);
- механізм LRU забезпечує більш високу ймовірність кеш-попадання в порівнянні з механізмом випадкового заміщення Random, однак цей виграш не дуже значний.

Відповідність між даними в оперативній пам'яті і в кеш-пам'яті забезпечується внесенням змін до тих області ОЗП, для яких дані в кеш-пам'яті зазнали змін. Існує два основних способи реалізації цих дій:

- з наскрізним записом (writethrough)
- зворотним записом (write-back).

При зчитуванні обидва способи працюють ідентично. При записуванні *кешування з наскрізним записом* оновлює основну пам'ять паралельно з оновленням інформації в КП. Це трохи знижує швидкодію системи, так як мікропроцесор згодом може знову звернутися за цією ж адресою для запису інформації, і попереднє пересилання рядка кеш-пам'яті в ОЗП виявиться марним. Однак при такому підході вміст відповідних один одному рядків ОЗП і КП завжди ідентичний. Це відіграє велику роль в мультипроцесорних системах із загальною оперативною пам'яттю.

Кешування зі зворотним записом модифікує рядок ОЗП лише при витісненні рядка кеш-пам'яті, наприклад, в разі необхідності звільнення місця для запису нового рядка з ОЗП в уже заповнену КП. Операції зворотного запису також ініціюються механізмом підтримки узгодженості кеш-пам'яті при роботі мультипроцесорній системи із загальною оперативною пам'яттю.

Проміжне становище між цими підходами займає спосіб, при якому всі рядки, призначені для передачі з КП в ОЗП, попередньо накопичуються в деякому буфері. Передача здійснюється або при витісненні рядка, як у випадку кешування зі зворотним записом, або при необхідності узгодження кеш-пам'яті декількох мікропроцесорів в мультипроцесорній системі, або при заповненні буфера. Така передача проводиться в пакетному режимі, що більш ефективно, ніж передача окремого рядка.

Організація внутрішньої кеш-пам'яті мікропроцесора

Внутрішній кеш 32-розрядного універсального мікропроцесора є загальним при зверненні як до команд, так і до даних. Звернення ведеться за фізичними адресами.

Кеш-пам'ять зазвичай реалізується у вигляді асоціативного ЗП, в якому для кожного рядка зберігаються додаткові відомості, звані тегом, або ознакою, в якості

якого виступає адресний код або його частина. Коли в АЗП подається адреса, з ним одночасно порівнюються всі теги.

Внутрішня кеш-пам'ять в процесорі i486 реалізує наскрізний запис. Починаючи з МП Pentium використовується наскрізний або зворотний запис. У зовнішній КП застосовується будь-який спосіб запису або їх комбінація.

Внутрішня кеш-пам'ять МП i486 має ємність 8 Кбайт і організована у вигляді 4-канальної асоціативної пам'яті. Це означає, що дані з якого-небудь рядка ОЗП можуть зберігатися в будь-якому з 4 рядків кеш-пам'яті.

КП складається з наступних блоків:

- блоку даних,
- блоку тегів,
- блоку достовірності і LRU.

Блок даних містить 8 Кбайт даних і команд. Він розділений на 4 масиви (напрямки), кожен з яких складається з 128 рядків. Рядок містить дані з 16 послідовних адрес пам'яті починаючи з адреси, кратної 16. Індекс масивів блоку даних, що складається з 7 біт, відповідає 4 рядкам КП, по одному з кожного масиву. Чотири рядки КП з одним і тим же індексом називаються множиною.

У блоці тегів є один тег довжиною 21 біт для кожного рядка даних в КП. Блок тегів також розділений на 4 масиви по 128 тегів. Тег містить старші 21 біт фізичної адреси даних, що знаходяться у відповідному рядку КП.

У блоці достовірності і LRU міститься по одному 7-розрядному значенню для кожного з 128 множин рядків КП: 4 біта достовірності (V) по одному на кожний рядок множини і 3 біти (B0 ... B2), керуючі механізмом LRU. Біти достовірності показують, чи містить рядок достовірні ($V = 1$) або недостовірні ($V = 0$) дані. При програмного очищення КП і апаратному скиданні процесора всі біти достовірності скидаються в 0.

Адресація кеш-пам'яті здійснюється шляхом поділу старших 28 біт фізичної адреси на 2 частини. Молодші 7 біт з цих розрядів (розряди 10 ... 4 фізичної адреси) утворюють поле індексу і визначають множину, в якому можуть зберігатися дані. Старші 21 біт (розряди 31 ... 11 фізичної адреси) служать полем тега і застосовуються

для визначення того, чи знаходиться інформація з даною фізичною адресою в будь-якої рядку обраної множини. Пошук в кеш-пам'яті інформації з заданою фізичною адресою виконується наступним чином:

Фізична адреса, за якою відбувається звернення, розбивається на 3 поля: Тег, Індекс, № байта. 7 розрядів $A10 \dots A4$ поля індексу визначають одне з 128 множин.

В обраній множині містяться 4 рядки з інформацією. Щоб визначити, чи присутня потрібна інформація в одному з рядків цієї множини, проводиться порівняння старших 21 біта фізичної адреси (поле Тег) з тегамі рядків обраної множини. Порівняння проводиться тільки для достовірних рядків, тобто тих, у яких в блоці достовірності встановлений біт достовірності $V = 1$.

Якщо для одного з рядків тег і розряди $A31 \dots A11$ фізичної адреси збіглися, то це означає, що сталося кеш-попадання і необхідна інформація є в кеш-пам'яті.

Зчитується знайдений рядок з 16 байт. Шуканий байт в ній визначається 4 молодшими розрядами фізичної адреси ($A3 \dots A0$).

Якщо на етапі 3 збігу не відбулося або всі рядки безлічі недостовірні, ця ситуація визначається як кеш-промах. В цьому випадку по сформованій мікропроцесором фізичній адресі виконується звернення до оперативної пам'яті. З ОЗП витягується потрібна інформація, і її рядок записується в вільний рядок обраної множини. Старші 21 біт фізичної адреси записуються в поле тега цього рядка. Якщо всі рядки в обраній множині достовірні, то заміщається рядок, до якого найдовше не було звернень згідно з механізмом LRU. Цей механізм діє точно так само, як і при витісненні рядків з буфера асоціативної трансляції TLB.

Режим роботи кеш-пам'яті визначається програмно установкою розрядів CD (заборона кешування) і NW (заборона наскрізного запису) в регістрі CR0. Кешування можна дозволити (це стан після ініціалізації при скиданні), можна заборонити при наявності достовірних рядків (в цьому режимі КП діє як швидке внутрішнє ОЗП) або, нарешті, кешування може бути повністю заборонено.

Управління роботою кеш-пам'яті на рівні сторінок

В елементах каталогу сторінок і таблиць сторінок є 2 біта, які застосовуються для управління вихідними сигналами процесора і беруть участь в кешуванні сторінок.

Біт PCD забороняє ($PCD = 1$) або дозволяє ($PCD = 0$) кешування сторінки. Заборона кешування необхідна для сторінок, які містять порти вводу/виводу з відображенням на пам'ять. Воно також корисно для сторінок, кешування яких не дає виграшу у швидкодії, наприклад, сайти можуть містити матеріали програми ініціалізації.

Біт PWT визначає метод оновлення ОЗП і зовнішньої кеш-пам'яті (кеш 2-го рівня). Якщо $PWT = 1$, то для даних у відповідній сторінці визначається кешування з наскрізним записом, при $PWT = 0$ застосовується спосіб зворотного запису. Використовується в мікропроцесорах починаючи з Pentium. Так як внутрішня кеш-пам'ять в МП i486 працює з наскрізним записом, стан біта PWT на неї не впливає. Біт PWT в цьому випадку діє тільки на зовнішню КП.

Забезпечення узгодженості кеш-пам'яті мікропроцесорів в мультипроцесорних системах

Розглянемо особливості роботи кеш-пам'яті в тому випадку, коли одночасно кілька мікропроцесорів використовують загальну оперативну пам'ять (рис. 4.3). В цьому випадку можуть виникнути проблеми, пов'язані з кешуванням інформації з оперативної пам'яті в кеш-пам'ять мікропроцесорів.

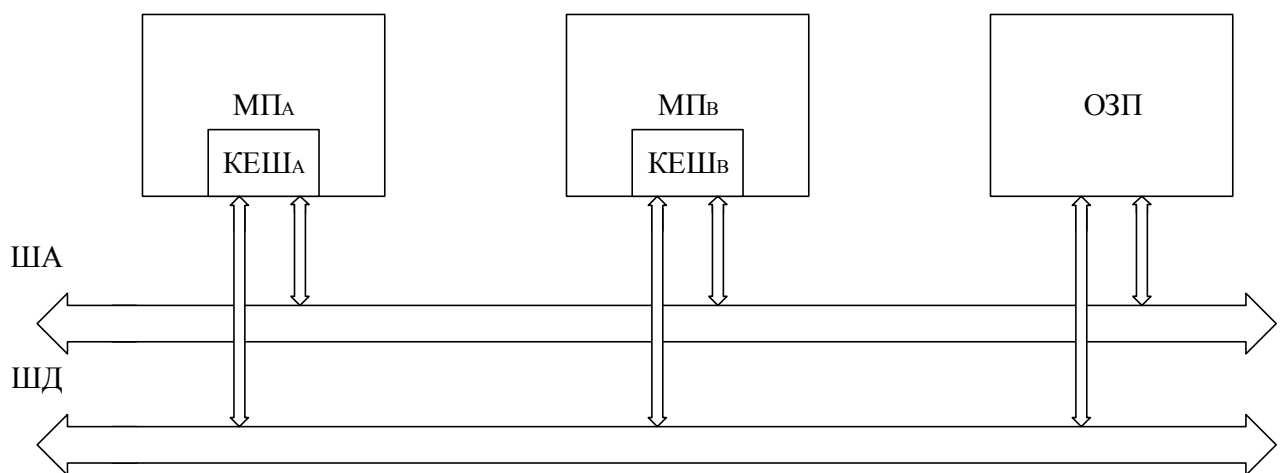


Рис. 4.3. Структура мультимікропроцесорної системи із загальною оперативною пам'яттю

Припустимо, що МП_А зчитав деякий рядок даних з ОЗУ в свою внутрішню КП і змінив дані в цьому рядку в процесі роботи. Ми відзначали, що існує два основних механізми оновлення оперативної пам'яті:

- наскрізний запис, при якому як тільки змінилася інформація у внутрішній кеш-пам'яті, ця ж інформація копіюється в те ж місце оперативної пам'яті;
- зворотний запис, при якому мікропроцесор після зміни інформації у внутрішньому кеші відображає цю зміну в оперативній пам'яті не відразу, а лише в той момент, коли відбувається витіснення цього рядка з кеш-пам'яті в оперативну.

Тобто існують певні моменти часу, коли інформація, припустимо, за адресою 2000 має різні значення: мікропроцесор її оновив, а в оперативній пам'яті залишилося старе значення. Якщо в цей момент інший мікропроцесор (МП_В), який використовує ту ж оперативну пам'ять, звернеться за адресою 2000 на ОЗУ, то він прочитає звідти стару інформацію, яка до цього часу вже не актуальна.

Для забезпечення узгодженості (когерентності) пам'яті в мультипроцесорних системах використовуються апаратні механізми, що дозволяють вирішити цю проблему. Такі механізми називаються протоколами когерентності кеш-пам'яті. Ці протоколи покликані гарантувати, що будь-яке зчитування елемента даних повертає останнє за часом записане в нього значення.

Існує два класи протоколів когерентності:

- протоколи на основі довідника (directory based): інформація про стан блоку фізичної пам'яті міститься тільки в одному місці, званому довідником (фізично довідник може бути розподілений по вузлах системи);
- протоколи спостереження (snooping): кожен кеш, який містить копію даних деякого блоку фізичної пам'яті, має також відповідну копію службової інформації про її стан; централізована система записів відсутня; зазвичай кеші розташовані на загальній шині, і контролери всіх кешей спостерігають за шиною (переглядають її), щоб визначати, які

звернення за адресами в межах цього блоку відбуваються з боку інших мікропроцесорів.

У мультипроцесорних системах із загальною пам'яттю найбільшою популярністю користуються протоколи спостереження, оскільки для опитування стану кешів вони можуть використовувати вже існуюче фізичне з'єднання - шину пам'яті.

Для підтримки когерентності застосовується два основні методи.

Один з методів полягає в тому, щоб гарантувати, що процесор повинен отримати виняткові права доступу до елементу даних перед виконанням запису в цей елемент даних. Цей тип протоколів називається протоколом запису з анулюванням (write invalidate protocol), оскільки при виконанні запису він анулює інші копії. Це найбільш часто використовуваний протокол як в схемах на основі довідників, так і в схемах спостереження. Виключне право доступу гарантує, що під час виконання запису не існує ніяких інших копій елемента даних, в які можна писати або з яких можна читати: всі інші кешовані копії елемента даних анульовані.

Альтернативою протоколу запису з анулюванням є оновлення всіх копій елемента даних в разі запису в цей елемент даних. Цей тип протоколу називається протоколом запису з оновленням (write update protocol), або протоколом запису з трансляцією (write broadcast protocol).

Ці дві схеми багато в чому схожі на схеми роботи кеш-пам'яті з наскрізним і зі зворотним записом. Ключовим моментом реалізації в багатопроцесорних системах з невеликим числом процесорів як схеми запису з анулюванням, так і схеми запису з оновленням даних, є використання для виконання цих операцій механізму шини. Для виконання операції оновлення або анулювання процесор просто захоплює шину і транслює по ній адресу, за якою має здійснюватися оновлення або анулювання даних. Всі процесори безперервно спостерігають за шиною, контролюючи адреси, які з'являються на ній. Процесори перевіряють, чи не знаходиться в їх кеш-пам'яті адреса, що з'явилася на шині. Якщо це так, то відповідні дані в кеші або анулюються, або оновлюються в залежності від використовуваного протоколу.

Розглянемо один з найбільш поширених протоколів, що забезпечують узгоджену роботу кеш-пам'яті декількох мікропроцесорів і основної пам'яті в мультимікропроцесорних системах, протокол MESI, який відноситься до групи протоколів спостереження з анулюванням. Будемо знайомитися з ним на прикладі двохпроцесорної системи, що складається з мікропроцесорів А і В.

Цей протокол використовує 4 ознаки стану рядка кеш-пам'яті мікропроцесора, за першими літерами яких і називається протокол:

- змінений стан (Modified): інформація, що зберігається в кеш-пам'яті мікропроцесора А, достовірна тільки в цьому кеші; вона відсутня в оперативній пам'яті і в кеш-пам'яті інших мікропроцесорів;
- виняткова копія (Exclusive): інформація, що міститься в кеші А, міститься ще тільки в оперативній пам'яті;
- колективна інформація (Shared): інформація, що міститься в кеші А, міститься в кеш-пам'яті принаймні ще одного МП, а також в оперативній пам'яті;
- недостовірна інформація (Invalid): в рядку кеш-пам'яті знаходиться недостовірна інформація.

Таким чином, стан ознак потоку MESI відображає наступні стани (по відношенню до МПА) рядки кеш-пам'яті (табл. 4.2):

Таблиця 4.2. Формування признаков стану протоколу MESI

Стан признаку протоколу	Стан рядка пам'яті		
	Кеш А	Кеш В	ОЗП
Modified	Д	НД	НД
Shared	Д	Д	Д
Exclusive	Д	НД	Д
Invalid	НД	Х	Х

При роботі мікропроцесора А з точки зору забезпечення когерентності пам'яті можливі наступні ситуації:

- RH (Read Hit) - кеш-попадання при читанні;
- WH (Write Hit) - кеш-попадання під час запису;
- RME (Read Miss Exclusive) - кеш-промах при читанні;
- RMS (Read Miss Shared) - кеш-промах при читанні, але відповідний блок є в кеш-пам'яті другого мікропроцесора;
- WM (Write Miss) - кеш-промах під час запису;
- SHR (Snoop Hit Read) - виявлення копії блоку при прослуховуванні операції читання іншого кешу;
- SHW (Snoop Hit Write) - виявлення копії блоку при прослуховуванні операції записи іншого кеша.

Найбільший інтерес тут представляють дві останні позиції. Сучасні мікропроцесори мають двосторонню шину адреси. Видаючи інформацію на цю шину, мікропроцесор адресує осередки оперативної пам'яті або пристрою введення-виведення. В силу того, що в даній мультипроцесорній системі мікропроцесори пов'язані спільною шиною, в тому числі і шиною адреси, приймаючи інформацію по адресним лініях, мікропроцесор визначає, чи було звернення за адресами, що містяться в його кеш-пам'яті, з боку інших мікропроцесорів. При виявленні такого звернення змінюється стан рядка кеш-пам'яті мікропроцесора.

Зміни ознаки стану блоку кеш-пам'яті МП в залежності від різних ситуацій в його роботі і роботі мультимікропроцесорної системи в цілому представлені на рис. 4.4.

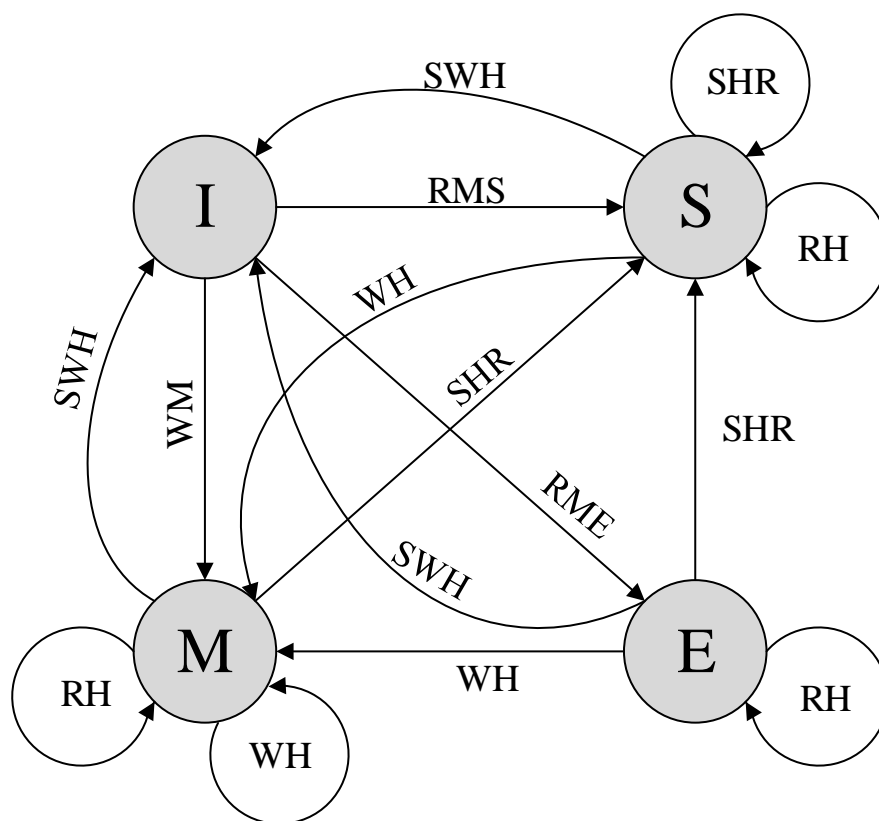


Рис. 4.5. MESI-діаграма забезпечення когерентності кеш-пам'яті

Проілюструємо деякі з представлених переходів.

Нехай блок кеш-пам'яті знаходиться в стані Modified, тобто достовірною інформацією знаходиться тільки в кеш-пам'яті даного МП. Тоді в разі виявлення при прослуховуванні адресної шини звернення з боку іншого мікропроцесора для читання інформації з якої входять у цю рядок адресами мікропроцесор повинен передати цей рядок кеш-пам'яті в ОЗУ, звідки вона вже буде прочитана іншим мікропроцесором.

При цьому стан рядка в кеш-пам'яті розглянутого мікропроцесора зміниться з модифікованого на колективне (Shared).

Якщо рядок кеш-пам'яті перебував в стані Invalid, тобто інформація в ньому була недостовірною, то по відношенню до цього рядку слід розглядати тільки ситуації, пов'язані з кеш-промахами. Так, якщо стався кеш-промах при виконанні операції запису, то необхідний рядок буде занесений в кеш-пам'ять даного МП, в цей рядок будуть записані змінені дані, і він набуде статусу виключного власника нової інформації (Modified).

У лекції розглянуті загальні принципи функціонування кеш-пам'яті мікропроцесора, організація кеш-пам'яті з прямим відображенням, повністю асоціативної і для множини асоціативної КП. Розглянуто основні механізми оновлення оперативної пам'яті: кешування з наскрізним і зі зворотним записом. Представлена організація внутрішньої кеш-пам'яті мікропроцесора. Розібрані способи забезпечення узгодженості кеш-пам'яті мікропроцесорів в мультипроцесорних системах.