Принципи на кеш паметите

Кеш паметите се използват като най-бързата памет и същевременно осигурява възможно най-голямо количество памет. Концепция за кеш е показана на фиг.3а. Има съответно голяма и бавна ОП. Кешът се състои от копия или блок от ОП. Когато процесорът се опита да прочете дума от паметта, проверка се прави дали думата е в кеша. И ако е така думата се праща на процесора. Ако не, то блокът с думата от ОП се зарежда в кеша. На фиг.3б е показано използването на кеш от три нива. Кешът от 2-ро ниво е по-бавен и по-голям от L1 и L3 е по-голям и по-бавен от L1. Нека да си преговорим структурата на кеш системата и ОП. ОП се състои от 2^n адресируеми клетки, като всяка клетка има уникален n–битов адрес. За маппинга е необходимо паметта да е разделена на блокове с фиксирана големина К клетки всеки. Следователно $M=2^n/K$ блокове в ОП. Кешът се състои от m блокове, наречени линии. Всяка линия се състои от К клетки и таг. Всяка линия се състои от управляващи битове(които не са показани), както и от битове които определят дали линията не е била модифицирана, след като е била заредена. Дължината на една линия като не се включват тага и контолните битове се нарича размер на линията. Има повече блокове отколкото в кеша. И етикета служеше за указване от кои блок е извлечена линията.

Следва механизма на извличане на блок от ОП. Операцията е четене. Последните две операции се извършват паралелно и оказват влияние върху организацията както е показано на фиг.4 – която е типична вече за съвременните кеш памети. При тази организация кеша е свързан с процесора чрез даннови, адресни и управляви сигнали. Данновите и адресните линии също са свързани с даннов и адресен буфер, който пък е свързан със системната шина, чрез която се достига ОП. Когато има кеш попадение дановия и адресния буфер стават неактивни и комуникацията е само между процесора и кеша без трафик по системната шина. Но при кеш miss търсения адрес се поставя на системната шина и данните се връщат през буфера за данни едновременно към кеша и процесора. При другите организации, кеша стои физически между процесора и основната памет и тогава данните първо се зареждаха в кеша и чак тогава се предаваха от кеша към процесора.

Елементи на кеша

1.Логически и физически кеш

Въпреки че проектирането на кешовете в HPC е по различно от другите платформи. Много разработчици на КА прилагат кешовете в простите компютърни архитектури. В повечето случаи има и ВП, която позволява програмите да адресират клетки от паметта от логична гл.точка – тогава адресното поле на машинните инструкции съдържат ВА-си. Тогава за да прочетат или запишат от ОП логиката за управление на паметта(MMU) преобразува всеки ВА в ФА на ОП. И тогава системният проектант трябва да избере къде да постави кеша – между процесора и MMU или м/у MMU и ОП фиг.4.7. Логическият кеш, познат като виртуален кеш, съхранява данните използвайки виртуални адреси. Процесорът достъпва кеша директно без да се обръща към MMU. А физическият кеш съхранява данните които имат ФАси на ОП. Едно от предимствата на логическия кеш е, че достъпа до него е по-бърз от физическия, защото кеша може да отговори преди да се представи MMU транслация. Недостатък е че повечето виртуални системи осигуряват едно и също виртуално адресно пространство за различните приложения. Това е така защото всяко приложение вижда ВП с начален адрес 0. И тогава 2 различни приложения се обръщат към едно и също място към ВП за различни физически адреси. Тогава кеша трябва да подържа режим за превключване м/у контекстите или да има допълнителни битове към всяка една линия за да указват към кое адресно пространство този адрес принадлежи. Има различни дискуси на тази тема.

2. Размер на кеша
Има редица възражения по повод увеличаване на кеша в КА – и не е само повече кеш по-скъпа конфигурация. Ако е по-голям кеша тогава ще имаме по-голям брои гейтове включени в адресирането на кеша. В резултат на което по-големия кеш ще бъде малко по-бавен от такъв с по-малък размер. Освен това мястото на чипа също е ограничено. Имаме оптимална големина на кеша, защото производителността на кеша зависи от естеството на работното множество. На таблицата са дадени стойности на кеша в някои по известни в миналото и днес КА.

3. Функция на съответствието
3.1. Директно съответствие
Тази техника е проста и не много скъпа за реализиране. Нейният основен недостатък е, че има фиксирано място в кеша за всеки блок. И ако в програмата има обръщение към клетки от различни блокове, но с един и същи адрес тогава ще се наложи премахването на едната линия и зареждане на новия блок. Но има един момент в които може да възникне ефекта trashing – тогава hit ratio намалява.
Слайт 11 –
Индексът с разпознат на две части: INDEX и WORD. Сравняването става по етикет – TAG общ за цялата линия от думи. Линия – това е съдържанието на една клетка в кеша и още линията е обема информация, която се прехвърля наведнъж между кеша и ОП. Чете се не 1 WORD , а цяла линия от WORD. Ако линията, съдържаща думата я няма в кеша се обръщаме към ОП
Основен недостатък на Директното съответствие е, че едноименни клетки от различни блокове на ОП се конкурират за една и съща клетка в кеша. Това означава, че ако се обръщаме към кеша, трябва да го освободим. Директното съответствие се осъществява много лесно, но вероятността да се окаже че кеш паметта е заета е много често срещано
3.2. Асоциативен метод на достъп
Покрива недостатъците от директното съответствие като позволява на всеки блок от ОП да бъде зареден едновременно в кеша. Но тук проблемът е следният когато зареждаме нов блок от ОП кои да заместим. Използват се различни алгоритми с цел увеличаване на hit . Основният недостатък на асоциативния метод на достъп е използването на сложните схеми за едновременно сравнение на всияки тагове на кеша.
Слайт 13 - Адреса се разглежда като етикет /TAG/ и клетка /WORD/. Блока от ОП се идентифицира етикета. Едновременно се анализират всички клетки на Кеша по адресната си част за съответствие с адреса на клетката от ОП. Ако има съвпадение получените данни се изпрящат към процесора,ако не се обръщаме към ОП. Имаме пълна свобода в разполагането на клетките от ОП в Кеша, което е много удобно, но много скъпо. Поради тази причина не се използва а се замества със схемата на Групово-асоциативното съответствие.
Слайт 13- Адреса на ОП се състои от 22 битов етикет /tag/ и 2 бита клетка /word/. Етикета трябва да се сравнява със 32 битовия блок от данни за всяка линия в кеша.
3.3. Групово-асоциативен кеш
Той е взаимства предимствата на двата подхода, за целта кеша се състои от групи, а всяка група се състои от линии.
фиг.6.
слайт 14 Примера
слайт 15 Индексът е адресация вътре в Кеша, но вътре в Кеша се адресират не само клетки, а група от клетки. В един SLOT на Кеша попадат няколко линии - имаме група от линии. Адресът се разделя на две части: етикет и индекс. По индекса в Кеша се адресира не само една клетка, а група от клетки т.е. няколко линии. Всяка от тези

клетки притежава свои собствени данни със съответния етикет. Клетка с даден индекс може да попадне на което и да е място в групата от клетки. Броят на групите може да бъде различен, но най-често се използват 2 или 4 групи.
фиг.7. на тази фигура са показани резултатите от симулацията на групово-асоциативен кеш с различен брои групи. От фиг.се вижда че разликата м/у директното съответствие и групово-асоц. е значителна докато размера на кеша е 64к