САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Архитектура ЭВМ

Отчет

по домашней работе № 3

**«КЭШ-ПАМЯТЬ»**

Выполнил: Ивченков Дмитрий Артемович

Номер ИСУ: 334906

студ. гр. M3134

Санкт-Петербург

2021

**Цель работы:** решение задач по теме «кэш-память».

**Теоретическая часть**

Кэш-память – это быстродействующая буферная память небольшого ограниченного объёма, расположенная близко к процессору, содержащая наиболее часто используемую информацию и предназначенная для ускорения обращения к ней. Кэш обладает существенно большей скоростью доступа, чем основная память, но и существенно меньшим объёмом по сравнению с ней.

Одним из значимых параметров кэш-памяти является *уровень ассоциативности*. Последовательный перебор всех линий кэша при поиске необходимых данных производился бы медленно и нивелировал бы всю пользу и смысл использования кэш-памяти. Возникает потребность в способе связать блоки данных оперативной памяти с кэш-линиями, поэтому ячейки памяти привязываются к кэш-линиям, т.е. в каждой линии могут быть данные из фиксированного набора адресов. Существует несколько возможных вариантов:

1. *Полностью ассоциативный кэш*, где любая линия памяти может быть сохранена в любую линию кэша.
2. *Кэш прямого отображения*, где каждая линия оперативной памяти соответствует только одной определённой кэш-линии. Тогда каждая строка памяти связана со строго определённой линией кэш-памяти, а каждой кэш-линии соответствует несколько определённых строк памяти.
3. *Частично-ассоциативный кэш*, где каждый блок данных памяти может быть отображён в какую-то линию из нескольких параллельных наборов линий кэша.

Кэш хранит копии данных из оперативной памяти, поэтому повторное обращение к ним происходит намного быстрее. Хранящаяся в кэше информация разделена на две части: сами данные и их адрес в системной памяти. Порции кэширования называются *кэш-линиями*. Адресное поле состоит из двух частей: *тега*, содержащего старшие биты адреса, и *индекса*, содержащего соответственно младшие биты. Тег является адресом расположения блока данных в основной памяти. Он также может содержать некоторую дополнительную информацию о статусе данных. Индекс же указывает на адрес кэш-линии в блоке и адрес байта в кэш-линии. Размер тега зависит от кэшируемого размера оперативной памяти , размера кэш-памяти , ассоциативности кэша и размера кэш-линии и может быть вычислен по формуле:

|  |  |
| --- | --- |
|  | (1) |

Тег является частью адреса памяти, в котором содержится информация (адрес) о расположении блока данных в основной памяти, о расположении кэш-линии в кэш-памяти и о расположении байта в кэш-линии. Количество бит адреса памяти равно двоичному логарифму из её размера. Адрес кэш-линии в кэше с какой-либо ассоциативностью ссылается «блок» из нескольких линий, их количество равно ассоциативности кэша, т.е. количество бит адреса кэш-линии равно двоичному логарифму из отношения количества кэш-линий в кэше к ассоциативности кэша. Наконец количество бит, необходимое для кодирования адреса байта внутри кэш-линии, равно двоичному логарифму из размера кэш-линии. Поэтому размер тега – это размер адреса памяти без адреса кэш-линии (блока из кэш-линий) и без адреса байта в кэш-линии, что показывает представленная выше формула 1.

При обращении процессора к данным прежде всего проверяется их наличие в кэш-памяти. Если необходимая информация находится в кэше, то происходит случай, называемый *попаданием кэша*(Hit), и она сразу используется. Иначе происходит *промах кэша*(Miss) и требуемые данные прочитываются из основной памяти, передаются в процессор и записываются в кэш, становясь доступными для следующих обращений. Процент обращений к кэшу, когда в нём найден результат, называется *коэффициентом попаданий*(), процент обращений, когда результат не найден и задействуется основная память, называется *коэффициентом промахов*(). *Время попадания*() – время, необходимое для получения данных из кэша в случае попадания кэша. *Штраф за промах* () – это время, необходимое для поиска и доставки данных с последующего уровня иерархии памяти после промаха данного кэша. *Среднее время обращения (доступа) к памяти*( – average memory access time) – это показатель для анализа производительности системы памяти. Он зависит от времени попадания, штрафа за промах и частоты промахов, может быть посчитан как в физических единицах времени, так и в тактах процессора. Этот параметр может быть рекурсивно расширен на несколько уровней иерархии памяти и вычисляется по формуле:

|  |  |
| --- | --- |
|  | (2) |

Может произойти промах кэша с вероятностью , тогда для получения данных дополнительно понадобится время штрафа за промах . С вероятностью происходит попадание кэша и весь процесс происходит только за . При расчёте взвешенного среднего нужно учитывать вероятность для каждого из слагаемых, но в данном случае в расчёт не принимается коэффициент попаданий. В общем случае предполагается, что, во-первых, вероятность промаха достаточно мала и значение близко к единице, а во-вторых, время попадания значительно меньше штрафа за промах , что ещё больше приближает значение множителя к единице.

Способы доступа к кэш-памяти могут быть разными. Существует два основных и широко распространённых: *look-through* и *look-aside*. При обращении с политикой доступа look-through контроллеру сначала нужно получить ответ от кэш-памяти, при попадании кэша прочесть из него необходимые данные и только при промахе кэша обратиться к следующему уровню иерархии памяти, дойдя в худшем случае до основной памяти. При работе с политикой look-aside запрос на чтение отправляется одновременно в кэш-память и в оперативную память. Если происходит попадание кэша, то запрос в основную память прерывается, а при промахе кэша необходимо дождаться ответа от памяти.

Кэш процессора разделён на несколько уровней. Самым быстрым, но и самым маленьким по объёму является кэш первого уровня – *L1*. Он разделён на две части: отдельно для команд (Instruction) и для данных (Data). Кэш второго уровня *L2* хранит больше информации, но его скорость также меньше. Кэш третьего уровня *L3* – самый большой и медленный кэш, но его скорость всё ещё существенно больше, чем у оперативной памяти, его объём масштабируется с количеством ядер. Кэши L1 и L2 собственные у каждого ядра, а L3 находится в общем пользовании и ведёт себя как единый уровень, хотя и состоит из отдельных блоков, соединённых кольцевой шиной.

**Практическая часть**

**Условие первой задачи:**

Имеется система с двухуровневым look through кэшем. Время отклика L1 и L2 равно 1 и 8 тактов соответственно. Штраф за промах из L2 в основную память равен 18 тактов. Коэффициент промахов для L2 в 2 раза меньше, чем для L1. Среднее время обращения к памяти (AMAT) равно 2 тактам.

Необходимо определить коэффициенты промахов для L1 и L2.

**Решение первой задачи:**

В системе с двухуровневым кэшем (с уровнями L1 и L2) используется политика доступа look-through, что даёт нам понять алгоритм работы. Когда процессор посылает запрос на чтение данных, сначала он обращается в кэш первого уровня. Если произошло попадание кэша L1, то он отвечает на запрос. Иначе произошёл промах и запрос посылается на уровень ниже, в кэш второго уровня. Снова, если требуемые данные нашлись в L2, то процессор получает ответ. Если же необходимых данных в кэше второго уровня нет, то остаётся только направить запрос на чтение в основную память и уже окончательно получить оттуда информацию. Таким образом, как говорилось ранее, для системы с двумя уровнями кэш-памяти формула 2 может быть расширена. Среднее время обращения к памяти у всей системы тогда будет равно сумме времени попадания L1 и коэффициента промаха L1, умноженного на среднее время обращения к памяти у кэша L2. Это равенство можно трактовать следующим образом: ответ может быть дан сразу из первого кэша за время HitTime1 или с вероятностью MissRate1 произойдёт промах первого кэша, тогда запрос будет послан во второй уровень кэша и потребуется его среднее время обращения к памяти AMAT2 (являющееся штрафом за промах MissPenalty1 первого кэша). А в случае обращения к второму кэшу произойдёт аналогичная ситуация: ответ может быть дан из второго кэша за время HitTime2 или с вероятностью MissRate2 произойдёт промах второго кэша, за которым последует обращение к основной памяти за время, равное штрафу за промах MissPenalty2 второго кэша.

|  |  |
| --- | --- |
|  | (3) |

Для решения задачи составим и решим уравнение. Необходимо определить коэффициенты промахов для L1 и L2, т.е. найти значения и .

Пусть – коэффициент промахов для L2. Из условия знаем:

Время отклика L1 и L2, т.е. время ответа на запрос, равно 1 такт и 8 тактов соответственно. Если произошло попадание, то спустя время отклика кэша процессор получил данные, т.е. , .

Штраф за промах из L2 в основную память равен 18 тактов.

Коэффициент промахов для L2 в 2 раза меньше, чем для L1. Значит, .

Среднее время обращения к памяти равно 2 тактам. .

Подставив известные величины в формулу 3, получим:

Очевидно, что значение не может быть ответом, так как коэффициент промахов неотрицателен. Значит, подходит значение . Отсюда получаем:

**Ответ:**

**Условие второй задачи:**

Имеется кэш с прямым отображением размером 32 КБ. Размер кэш-линии составляет 32 байта. Разрядность адресов памяти 32 бита.

Необходимо определить размер тега адреса.

**Решение второй задачи:**

Адрес памяти в кэше состоит из адреса блока данных в основной памяти, т.е. тега, адреса кэш-линии в кэше и адреса байта в кэш-линии. Тег адреса содержит старшие биты адреса памяти, поэтому размер тега адреса – это размер части полного адреса без адреса кэш-линии и адреса байта. Если кэшируемый размер оперативной памяти равен байт, то необходимо бит для кодирования полного адреса байта в основной памяти. В кэше с прямым отображением каждая линия из основной памяти может быть отображена только в определённую кэш-линию (прямое отображение можно ещё назвать ассоциативностью-1), поэтому в адресе используется число бит, нужное для кодирования номеров всех кэш-линий. Таким образом, по адресу можно однозначно найти заданную кэш-линию. Тогда если число линий в кэш-памяти равно , то требуется бит, чтобы закодировать адреса всех таких линий. Если размер кэш-линии равен байт, то понадобится бит для адреса каждого байта в кэш-линии. Тогда размер в битах тега адреса для кэша с прямым отображением можем вычислить по формуле 1.

По условию дано:

Разрядность адресов памяти равна 32 битам, т.е. сразу известен размер адреса памяти .

Размер кэша равен 32 КБ, размер кэш-линии . Разделив общий объём кэша на размер одной линии, получим количество линий в кэше . Как было сказано, адрес строки в кэше с прямым отображением ссылается на единственную строку ().

Подставим в формулу 1 известные размеры и вычислим размер тега:

**Ответ:**