

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО

Дисциплина: Архитектура ЭВМ

Отчет

по домашней работе №3

Кэш-память

Выполнил(а): Левицкий Иван Михайлович

Номер ИСУ: 334916

студ. гр. М3135

Санкт-Петербург

2021

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

Теоретическая часть

Разберемся со всеми терминами и зависимостями необходимыми для полного понимания условий и последующего решения предложенных задач.

Кэш и его типы:

Кэш - промежуточный накопительный буфер системы, хранящий в себе информацию и способный предоставить доступ к ней быстрее чем обычная оперативная память (или другой удаленный источник данных).

Кэш-память бывает многоуровневой и обычно имеет один, два или три уровня, каждый из которых обычно больше и медленнее предыдущего (уровень кэша обычно обозначается записью LN , где N - номер уровня кэша, например, $L1$ - кэш первого уровня).

Также кэш-память бывает разделенной внутри (Гарвардская архитектура) на отдельные кэши под команды и данные ($I(\text{instruction})$ - кэш под команды; $D(\text{data})$ - кэш под данные; тип кэша обычно указывается после номера уровня, например, $L1I$ - кэш первого уровня команд).

Ассоциативность и тэги адреса:

Полностью ассоциативная кэш-память (ассоциативность 0) - простейшая модель кэш-памяти в которой любая строка оперативной памяти может быть записана в любую строку кэш-памяти.

Кэш-память с прямым отображением (ассоциативность 1) - другая модель кэш-памяти, в которой наоборот, каждой строке оперативной памяти соответствует ровно одна строка кэш-памяти.

Наборно-ассоциативный кэш - кэш разделенный на несколько банков (сегментов), каждый из которых представляет собой кэш с прямым отображением, а между собой они полностью ассоциативны (заметим, что ассоциативностью кэша тогда называют кол-во банков в наборно-ассоциативном кэше).

Кэш-линия - несколько подряд записанных байтов (синонимично “строке кэш-памяти”).

Тэг (адрес сохраняемого байта) - код хранящий в себе всю основную информацию о байте, то есть его полный адрес, исключая информацию о номере его кэш-линии и его номере в этой кэш линии. Исходя из того определения, несложно вывести формулу размера тэга для кэш-памяти с прямым отображением:

$$S(Teg) = \log_2(Q(Mem)) - \log_2(Q(cache - lines)) - \log_2(S(cache - line))$$

где $S(Teg)$ - искомая длина тэга, $Q(Mem)$ - объем всей памяти (кол-во всех полных адресов), $Q(cache - lines)$ - кол-во кэш-линий, $S(cache - line)$ - длина кэш-линии. Она верна т.к. $\log_2(Q(Mem))$ - длина всего адреса; $\log_2(Q(cache - lines))$ - длина кода, отвечающего за номер кэш-линии; $\log_2(S(cache - line))$ - длина кода, отвечающего за номер в самой кэш-линии. Единственным уточнением к этой формуле будет то, что длина кода для кодирования всех кэш-линий для наборно-ассоциативного кэша на самом деле будет меньше, т.к. мы не будем хранить информацию о том в

какой именно банк (сегмент) попадает информация, а значит вместо $\log_2(Q(cache - lines))$ в формулу можно подставлять $\log_2(Q(cache - lines)/A)$, где A - ассоциативность кэша, т.е. нашу формулу можно обобщить для наборно-ассоциативных кэшей (учитывая св-во разности логарифмов):

$$S(Teg) = \log_2(Q(Mem)) - \log_2(Q(cache - lines)) + \log_2(A) - \log_2(S(cache - li$$

Другими словами, в тэг будет включен код, отвечающий за номер банка.

Измерение среднего времени обращения к памяти:

AMAT (Average memory access time) - среднее время обращения к памяти (измеряющееся либо в тактах ЦП, либо в наносекундах).

$$AMAT = HR * H + MR * (AMP + H)$$

- формула вычисляющая значение AMAT для одноуровневой кэш-памяти (где, H (hit latency) - время, расходуемое на обращение к памяти при поп

адании в кэш (может сокращенно называться задержкой попадания/или временем отклика); MR (miss rate) - коэффициент промаха, т.е. вероятность того, что строки памяти к которой мы обращаемся не окажется в кэш-памяти; HR (- коэффициент попадания (то есть величина $(1 - MR)$); AMP (average miss penalty) - штраф за промах, т.е. время которое потратит процессор если запрашиваемой строки памяти не окажется в кэш-памяти и он обратится напрямую к оперативной памяти). Формула достаточно очевидна и основана на принципе среднего взвешенного т.е. мы берем вероятность того, что попадем в память (HR) умножаем на время, которое

затратим при попадании (H), и складываем с временем, которое мы затратим при промахе (AMP) с учетом потраченного времени H на сам запрос, умноженным на вероятность промаха (MR), получая ответ в виде среднего времени обращения к памяти.

Преобразуя подобные, получаем формулу:

$$AMAT = H + MR * AMP$$

Практическая часть

Условие задачи №3

Имеются две системы, различающихся лишь кэшем.

На первой установлен общий кэш для команд и данных размеров 32 КБ. На второй используется отдельный кэш: 16 КБ для команд и 16 КБ для данных.

Штраф за промах 50 тактов. Коэффициенты промахов для кэшей представлены в таблице 1. Время отклика 1 такт для отдельных кэшей и доступа к командам в общем кэше, 2 такта для доступа к данным в общем кэше. 75% обращений от общего числа обращений к кэшу проводится по командам и 25% – по данным.

Таблица 1 – Коэффициент промахов для кэшей

Размер	Кэш команд	Кэш данных	Общий кэш
16 КБ	0.64%	6.47%	2.87%
32 КБ	0.15%	4.82%	1.99%

Нужно определить среднее время обращения к памяти (АМАТ) для этих кэшей.

Решение задачи №3

Формализуем нужные нам условия:

$AMP = 50$ тактов - общее значение для обеих систем и по командам и по данным.

$H_2 = H_{1I} = 1$ такт - время отклика для отдельных кэшей (2 система) и кэша команд (1 система) соответственно.

$H_{1D} = 2$ такта - время отклика для кэша команд (1 система).

$MR_1 = 1.99\% = 0.0199$ - общая вероятность промаха для кэша первой системы (отдельные нам не понадобятся, т.к. кэш общий).

$MR_{2I} = 0.64\% = 0.0064$ - вероятность промаха для кэша команд второй системы

$MR_{2D} = 6.47\% = 0.0647$ - вероятность промаха для кэша данных второй системы.

$P_I = 75\% = 0.75$ - вероятность (частота) запроса команд.

$P_D = 25\% = 0.25$ - вероятность (частота) запроса данных.

Посчитаем АМАТ для каждой системы отдельно (сначала для первой):

1)Посчитаем отдельно АМАТ для команд и данных а потом применим к ним формулу среднего взвешенного, зная вероятность (частоту) обращения ($AMAT = P_I * AMAT_I + P_D * AMAT_D$).

$$AMAT_{1I} = H_{1I} + MR_1 * AMP = 1 + 0.0199 * 50 = 1.995$$

$$AMAT_{1D} = H_{1D} + MR_1 * AMP = 2 + 0.0199 * 50 = 2.995$$

$$AMAT_1 = 0.75 * 1.995 + 0.25 * 2.995 = 2.245$$

2)Аналогично посчитаем отдельно АМАТ для команд и данных на второй системе, а потом применим формулу среднего взвешенного.

$$AMAT_{2I} = H_2 + MR_{2I} * AMP = 1 + 0.0064 * 50 = 1.32$$

$$AMAT_{2D} = H_2 + MR_{2D} * AMP = 1 + 0.0647 * 50 = 4.235$$

$$AMAT_2 = 0.75 * 1.32 + 0.25 * 4.235 = 2.04875$$

Ответ: $AMAT_1 = 2.245$ тактов, $AMAT_2 = 2.04875$ тактов

Условие задачи №7

Дана система, имеющая L1I (кэш первого уровня команд), L1D (кэш первого уровня данных) и L2 (кэш второго уровня) характеристиками, представленными в таблице 2.

Таблица 2 – Характеристики кэшей задачи 7

	Объём	Ассоциативность	Размер кэш-линии
L1I	4КБ	Прямое отображение	4 байта
L1D	4КБ	2	4 байта
L2	64КБ	4	16 байт

Разрядность адресов памяти 30 бит.

Необходимо определить размер тэгов адреса для всех трёх кэшей.

Решение задачи №7

Формализуем нужные нам условия:

Нам сразу дано, что $\log_2(Q(Mem)) = 30$, т.к. разрядность адреса памяти и есть эта величина.

для L1I: $V(Mem) = 4\text{КБ}$ (объем памяти кэша),
 $S(cache - line) = 4 \text{ байта}$, $A = 1$. Очевидно, что
 $Q(cache - lines) = V(Mem)/S(cache - line) = 2^{10}$, т.е. кол-во кэш
линий равно объему всей памяти, деленному на длину кэш-линии.

для L1D: $V(Mem) = 4\text{КБ}$, $S(cache - line) = 4 \text{ байта}$, $A = 2$, тогда по
анalogии подсчитаем $Q(cache - lines) = 2^{10}$.

для L2: $V(Mem) = 64\text{КБ}$, $S(cache - line) = 16 \text{ байтов}$, $A = 4$, тогда
по аналогии подсчитаем $Q(cache - lines) = 2^{12}$.

Остается все подставить в формулу:

$$S(Teg) = \log_2(Q(Mem)) - \log_2(Q(cache - lines)) + \log_2(A) - \log_2(S(cache - line))$$

L1I:

$$S(Teg) = 30 - \log_2(2^{10}) + \log_2(1) - \log_2(4) = 30 - 10 + 0 - 2 = 18$$

L1D:

$$S(Teg) = 30 - \log_2(2^{10}) + \log_2(2) - \log_2(4) = 30 - 10 + 1 - 2 = 19$$

L2:

$$S(Teg) = 30 - \log_2(2^{12}) + \log_2(4) - \log_2(16) = 30 - 12 + 2 - 4 = 16$$

Ответ: 18 бит, 19 бит и 16 бит, соответственно для L1I, L1D и L2.