

ИЗМЕРЕНИЕ СТЕПЕНИ АССОЦИАТИВНОСТИ КЭШ-ПАМЯТИ

Для экспериментального определения степени ассоциативности кэш-памяти в программе организуется обход данных в памяти, который вызывает «буксование» кэш-памяти. Для этого предлагается организовать доступ в память по адресам, отображаемым на одно и то же множество в кэш-памяти. Будем увеличивать количество таких конфликтующих обращений. Когда оно превысит степень ассоциативности кэш-памяти, то данные начнут вытесняться из кэш-памяти, и возникнет кэш-буксование. Буксование кэш-памяти приводит к увеличению времени выполнения программы, благодаря чему можно определить степень ассоциативности кэш-памяти.



рис. 1. Пример множественно-ассоциативной кэш-памяти

Например, для кэш-памяти, представленной на рис. 1, размер банка составляет 4 Кб, а степень ассоциативности равна четырем. Таким образом, если многократно выполнять обращения по пяти и более адресам, отстоящим друг от друга на расстояние, кратное 4 Кб, то мы получим эффект кэш-буксования.

Рассмотрим, как может быть организован обход памяти в соответствии с описанной идеей эксперимента. Будем выполнять обход нескольких фрагментов данных в памяти, отстоящих друг от друга на смещение, кратное размеру банка. Суммарный объем фрагментов данных возьмем равным размеру исследуемой кэш-памяти, предполагая, что для всех запрашиваемых данных в кэш-памяти должно найтись место.

Чтобы адреса отображались на одно и то же множество в кэш-памяти, необходимо, чтобы смещение между ними было кратно размеру банка кэш-памяти. Так как размер банка кэш-памяти не всегда известен, то рассмотрим, как его можно оценить. Размер кэш-памяти кратен размеру банка, поэтому, если известен размер кэш-памяти, то расстояние между началами фрагментов можно взять равным ему. Если же размер кэш-памяти неизвестен, то можно использовать тот факт, что размеры банков кэш-памяти практически всегда являются степенью двойки. Для современных процессоров подходящим смещением между фрагментами будет $2^{24} \text{ b} = 16 \text{ Mb}$.

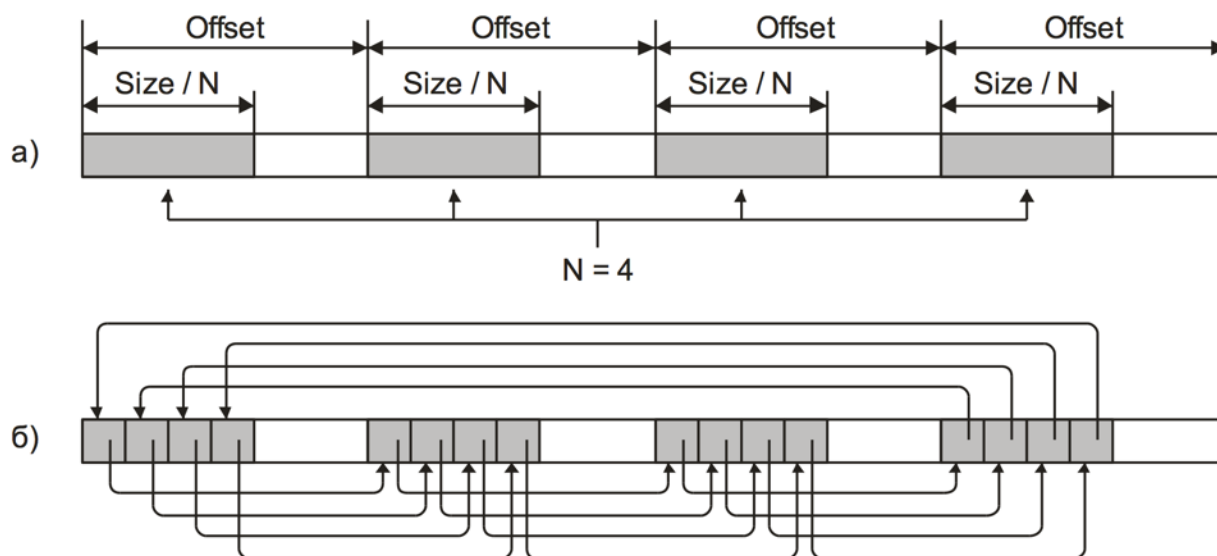


Рис. 2. Схема расположения в памяти фрагментов данных для обхода (а) и порядок обхода элементов (б)

Обход фрагментов данных организуем таким образом, чтобы подряд происходили обращения к элементам разных фрагментов, отстоящим на заданное смещение. Сначала последовательно производится чтение всех первых элементов во всех фрагментах, затем всех вторых, затем – всех третьих и т.д. На рис. 2 показано размещение фрагментов данных в памяти (рис. 2а) и порядок обхода элементов (рис. 2б). Здесь Size – объем кэш-памяти, N – число фрагментов, Offset – смещение между началами соседних фрагментов.

Для проведения эксперимента предлагается выделить массив достаточно большого размера, чтобы вместить все необходимые фрагменты с требуемым смещением. Элементы массива образуют связанный список, где значение каждого элемента является индексом следующего.

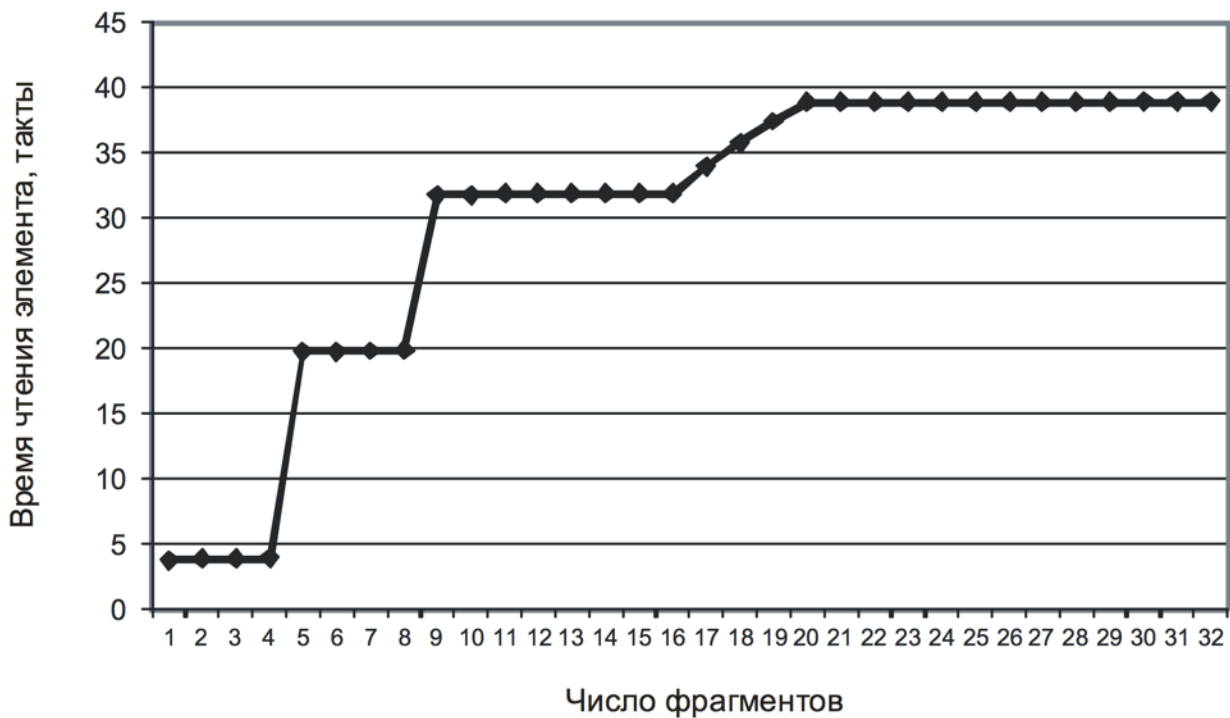


Рис. 3. Среднее время чтения одного элемента массива в зависимости от числа фрагментов для процессора Intel Xeon X5660.

Пример графика, полученного на процессоре Intel Xeon X5660 (степень ассоциативности кэш-памяти 1-го и 2-го уровня: 8, степень ассоциативности кэш-памяти 3-го уровня: 16), представлен на рис. 3. На графике видно замедление после 8-ми фрагментов, что соответствует степени ассоциативности кэш-памяти 1-го и 2-го уровня, а также замедление после 16-ти фрагментов, что соответствует степени ассоциативности кэш-памяти 3-го уровня. Увеличение времени после 4-х фрагментов тоже соответствует степени ассоциативности, но уже не кэш-памяти, а буфера трансляции адресов (TLB).

ЗАДАНИЕ

1. Написать программу, выполняющую обход памяти в соответствии с заданием.

2. Измерить среднее время доступа к одному элементу массива (в тактах процессора) для разного числа фрагментов: от 1 до 32. Построить график зависимости времени от числа фрагментов.

3. По полученному графику определить степень ассоциативности кэш-памяти, сравнить с реальными характеристиками исследуемого процессора.