

ЗАДАНИЕ И КОММЕНТАРИЙ К ПРАКТИЧЕСКОЙ РАБОТЕ №7 «ИЗМЕРЕНИЕ СТЕПЕНИ АССОЦИАТИВНОСТИ КЭШ-ПАМЯТИ»

В данной теме нужно хорошо разобраться с видами кэшей: кэш прямого отображения, полностью ассоциативный и множественно-ассоциативный (он же наборно-ассоциативный, он же частично-ассоциативный).

Современные процессорные кэши являются множественно-ассоциативными (почему выбирается именно такой тип будете объяснять мне вы). Устроены такие кэши по принципу таблицы (рис. 1 из описания лабы №9). Строками таблицы являются множества (наборы). Столбцами – банки ассоциативности. Количество банков ассоциативности – это степень ассоциативности. Она может быть различной для кэшей разных уровней вашего процессора. В ячейках таблицы находятся кэш-строки. Типичные размеры кэш-строк – 32 или 64 байта. Вам необходимо помнить о том, что кэш-строка является минимальной единицей циркуляции информации между оперативной памятью и процессорным кэшем. То есть считать из памяти меньше, чем кэш-строка, нельзя. Не следует путать понятие кэш-строка со строками таблицы, в виде которой представим кэш. Эти строки мы называем множествами.

Принцип заполнения множественно-ассоциативного кэша при последовательном чтении из памяти следующий. Будем полагать, что кэш обладает теми характеристиками, которые представлены на рис.1 описания к лабе 9. Из памяти считывается первые 32 байта, составляющие кэш-строку, и записываются в какую-то ячейку из 1-го множества. Затем считываются следующие 32 байта и заполняется какая-либо ячейка второго множества. После считывания объема памяти, равного 4 Кб, то есть когда мы пройдем все множества подряд, снова заполняется какая-то кэш-строка из первого множества, но только не та, что была заполнена в ходе записи первых 4 Кб.

То есть в ходе такого процесса заполнения очередная кэш-строка у заполнится в том же множестве, что и кэш-строка x , если между x и y расстояние в памяти = 4 Кб или 8 Кб или 12 Кб. А вот при обращении к данным, отстоящим на расстоянии 16Кб от первой порции размером 32Б, попавшей в это множество, для них уже не будет места в данном множестве (все кэш-строки множества заполнены) и придется вытеснять самые старые данные, находящиеся в данном множестве.

На этом принципе и основан алгоритм определения степени ассоциативности кэш-памяти. Мы всегда читаем данные, которые должны находиться в одном и том же множестве. Это обеспечивается тем, что мы читаем элементы массива, отстоящие друг от друга на одно и то же расстояние, кратное размеру банка ассоциативности. Если не знаем этот размер, то самое простое- шагать по массиву через размер кэша. Тогда при количестве фрагментов, меньшем либо равном степени ассоциативности, они будут уместиться во множество кэш-памяти, а когда количество фрагментов будет увеличиваться, то возникнет эффект «кэш-букования». Будут происходить обращения к оперативной памяти, а, следовательно, резко возрастет среднее время считывания элемента массива.

Среднее время доступа к элементу определяйте также, как в предыдущей практической работе:

```
for (...)
```

```
k=A[k];
```

Массив *A* должен представлять собой связный циклический список, идущий по элементам фрагментов так, как изложено в описании к лабе.

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.
4. Составить отчет по практической работе. Отчет должен содержать следующее.
 - a. Титульный лист.
 - b. Цель практической работы.
 - c. Параметры теста: размер фрагментов, величина смещения.
 - d. График зависимости среднего времени доступа к элементу массива от числа фрагментов.
 - e. Оценку степени ассоциативности различных уровней кэш-памяти согласно выполненным вычислительным экспериментам.
 - f. Реальные значения степеней ассоциативности различных уровней кэш-памяти процессора, подкрепленные доказательствами (скриншоты из программ типа CPU-Z, файлы операционной системы, куски официальной документации по процессору и т.д.)
 - g. Полный компилируемый листинг реализованной программы и команды для ее компиляции.
 - h. Вывод по результатам практической работы.