

# Методика оптимизации и оценки эффективности кэш-памяти второго уровня

С.И. Аряшев, С.Г. Бобков, П.В. Саяпин

Научно-исследовательский институт системных исследований РАН, [sayapin@cs.niisi.ras.ru](mailto:sayapin@cs.niisi.ras.ru)

**Аннотация** — Предложена методика оценки производительности микропроцессора в зависимости от оптимизации подсистемы памяти блоками буферизации. Описаны буфер входных данных и буфер сохранения данных кэш-памяти второго уровня. Проведенные на разрабатываемых в НИИСИ РАН микропроцессорах измерения показали, что блоки буферизации данных позволяют увеличить производительность микропроцессора более чем на 17%.

**Ключевые слова** — подсистема памяти, кэш-память, блок буферизации, команда сохранения данных, помехоустойчивое кодирование.

## I. ВВЕДЕНИЕ

В современных микропроцессорных системах критическим местом с точки зрения производительности является доступ к оперативной памяти. Это обусловлено тем, что, как правило, тактовая частота процессорного ядра в несколько раз выше тактовой частоты памяти, а микросхема может содержать число ядер, превышающее в несколько раз число каналов памяти. Даже в стандартных программах доля инструкций загрузки и сохранения данных может составлять более 30 % от общего количества инструкций. Например, в системе тестов SPECCPU2000 для целочисленного набора тестов количество команд загрузки/сохранения составляет 33,2%, для тестов с плавающей точкой - 32,6 % [1]. Для суперскалярных процессоров, выполняющих две инструкции за такт, требуется более 60 % инструкций загрузки и сохранения, а при выполнении трёх инструкций за такт приближаемся к необходимости обращения к памяти в каждом такте.

Еще более серьезные требования как к производительности, так и к надежности подсистемы памяти, возникают при разработке суперЭВМ для проведения полномасштабных расчетов сложных технических и биологических систем с учетом атомно-молекулярных взаимодействий. Это объясняется большими объемами входных данных и условий расчёта.

В современных компьютерах используется многоуровневая подсистема памяти. Более быстрые компоненты памяти имеют меньший объем. Быстрая SRAM память в современных микропроцессорах, как правило, находится на одном кристалле с процессорным ядром и называется кэш-памятью. Время доступа также зависит и от физического

расстояния, поэтому более быстрые компоненты располагают ближе к процессору. В результате типичная иерархия подсистемы памяти такова (в порядке удаления от процесса): регистры процессора, накристалльная кэш-память, оперативная память и виртуальная память [5]. Кэш-память также разделяется на несколько уровней, более высокий уровень имеет больший размер и, соответственно, большее время доступа. В современных микропроцессорах количество уровней кэш-памяти доходит до трёх.

Важную роль в повышении производительности многоуровневой подсистемы памяти играет буферизация запросов.

## II. ПОДСИСТЕМА ПАМЯТИ

В большинстве современных микропроцессоров для кэш-памяти данных первого уровня (КД1) в качестве политики записи используется сквозная запись (write-through), а для кэш-памяти второго уровня (К2) – обратная запись (writeback). Например, данная конфигурация используется в процессорах Bulldozer фирмы AMD [6]. Применение сквозной записи оправдано для кэш-памятей небольшого объема. В этом случае уменьшение нагрузки на шину данных при обмене с более низким уровнем подсистемы памяти осуществляется путем введения буфера записи. При этом уменьшение количества запросов будет большим, чем при использовании кэш-памяти с политикой обратной записи равного объема. С увеличением объема кэш-памяти, использующей в качестве политики записи обратную запись, значительно уменьшается количество запросов на запись в более низкий уровень подсистемы памяти [2].

В кэш-памятях, использующих сквозную запись, помехоустойчивость обеспечивается с помощью использования проверки битов четности. Так как при сквозной записи актуальные данные находятся как в кэш-памяти, так и на более низком уровне подсистемы памяти, то при возникновении ошибки при чтении данных фиксируется промах и запрос отправляется на более низкий уровень. При использовании обратной записи актуальные данные находятся только в кэш-памяти. В связи с этим для данных кэш-памятей вводится код коррекции ошибки, позволяющий определить факт сбоя и в некоторых случаях позволяющий исправить ошибку. В тех случаях, когда

исправление ошибки не представляется возможным, вызывается исключение.

Для данной конфигурации подсистемы памяти (для КД1 используется сквозная запись, для К2 – обратная запись) запросы на сохранение разделяются на четыре типа в зависимости от того, было ли попадание в кэш-память.

В ситуациях промаха в кэш-память второго уровня частота следования инструкций сохранения в значительной степени ограничена задержками записи в оперативную память. В случае попадания в К2 частота следования инструкций загрузки ограничена задержкой доступа к кэш-памяти второго уровня. Это обусловлено большим временем доступа к кэш-памяти второго уровня относительно кэш-памяти данных первого уровня. Задержка возникает из-за значительной разницы в объемах кэш-памяти первого и второго уровней. Как правило, объем К2 в несколько раз превышает объем КД1 (в данной статье рассматривается КД1 объемом 16 Кб и К2 объемом 256 Кб). Так как на простейших операциях копирования, как будет показано ниже, процент попаданий в кэш-память второго уровня от команд сохранения может достигать 75%, уменьшение задержек доступа к К2 позволяет существенно увеличить производительность системы. Уменьшение задержек доступа к кэш-памяти второго уровня осуществляется путем введения специальных аппаратных блоков для буферизации входных данных.

### III. ОЦЕНКА ОБЩЕГО УСКОРЕНИЯ СИСТЕМЫ В ЗАВИСИМОСТИ ОТ УСКОРЕНИЯ ОТДЕЛЬНЫХ ОПЕРАЦИЙ

Согласно закону Амдаля, улучшение производительности системы ограничено временем, в течение которого данное улучшение используется [3]. Отсюда *ускорение* определяется как:

$$\text{Ускорение} = \frac{Q_{\text{ул}}}{Q_{\text{исх}}}, \quad (1)$$

где  $Q_{\text{ул}}$  – производительность системы, с введённым в нее улучшением,  $Q_{\text{исх}}$  – исходная производительность системы.

С другой стороны,

$$\text{Ускорение} = \frac{T_{\text{исх}}}{T_{\text{ул}}},$$

где  $T_{\text{ул}}$  – время выполнения задачи при введении улучшения,  $T_{\text{исх}}$  – исходное время выполнения задачи.

В [4] было представлено следующее определение закона Амдаля:

$$\text{Ускорение} = \frac{1}{r_s + \frac{r_p}{n}},$$

где  $r_s + r_p = 1$ ,  $r_s$  – часть последовательного кода программы и  $n$  – количество потоков выполнения.

Для более точной оценки ускорения на операциях загрузки/сохранения в данной статье выводится и используется представление закона Амдаля, учитывающее влияние таких параметров, как количество тактов на инструкцию для команды сохранения и частота появления данной команды.

Количество тактов, затраченное на выполнение программы, определяется формулой:

Количество тактов =  $\sum_{i=1}^n IC_i \times CPI_i$ ,  
где  $IC_i$  – количество выполнений инструкции  $i$ ,  $CPI_i$  – среднее число тактов, затраченное на выполнение инструкции  $i$ .

Отсюда:

$$CPI_{\text{общ}} = \sum_{i=1}^n \frac{IC_i}{\text{Количествоинструкций}} \times CPI_i.$$

Следовательно, общее время выполнение программы (Т) может быть выражено формулой:

$$T = \text{Количество тактов} \times \text{Время такта} = (\sum_{i=1}^n IC_i \times CPI_i) \times \text{Время такта}.$$

Или, если выразить через  $CPI_{\text{общ}}$ :

$$T = \text{Количество инструкций} \times CPI_{\text{общ}} \times \text{Время такта}.$$

Тогда ускорение можно представить в виде:

$$\text{Ускорение} = \frac{\text{Количество инструкций} \times CPI_{\text{общ}} \times \text{Время такта}}{\text{Количество инструкций} \times CPI_{\text{общ}}^* \times \text{Время такта}} = \frac{CPI_{\text{общ}}}{CPI_{\text{общ}}^*},$$

где  $CPI_{\text{общ}}^*$  – среднее количество тактов на инструкцию при введении улучшения.

Связь  $CPI_{\text{общ}}$  и  $CPI_{\text{общ}}^*$  можно выразить следующим образом:

$$CPI_{\text{общ}}^* = CPI_{\text{общ}} - I \times (CPI_k - CPI_k^*),$$

где  $I$  – частота появления инструкции, для которой производилось улучшение;  $CPI_k$  – исходное количество тактов на инструкцию для данной инструкции;  $CPI_k^*$  – количество тактов на инструкцию для данной инструкции после ускорения.

Таким образом:

$$\text{Ускорение} = \frac{1}{1 - I \times \frac{CPI_k - CPI_k^*}{CPI_{\text{общ}}}}. \quad (2)$$

Из данной формулы следует, что зная начальный  $CPI$  и частоту появления инструкции, для которой производится оптимизация, можно оценить вклад предлагаемого решения в общее ускорение системы. Также из формулы видно, что ускорение зависит от начального состояния системы (значение  $CPI$  в отсутствие улучшений). Это значение учитывает остальные параметры системы, не зависящие от той операции, для которой производилось улучшение. Учитывая данный факт, оценка ускорения должна производиться для двух разных значений  $CPI_{\text{общ}}$ .

#### IV. ФУНКЦИОНИРОВАНИЕ КЭШ-ПАМЯТИ ВТОРОГО УРОВНЯ

Кэш-память второго уровня состоит из блоков памяти данных, тэгов и проверочных битов, а также из контроллера кэш-памяти.

подавляющее большинство современных микропроцессоров используют наборно-ассоциативную организацию кэш-памятей. При наборно-ассоциативной организации блок памяти может размещаться в ограниченном диапазоне мест в кэш-памяти. Вначале определяется секция для размещения блока (секцией является группа блоков в кэш-памяти). Далее блок может быть размещен в любом месте внутри данной секции [3].

Физический адрес в кэш-памяти логически разделяется на две части: адрес блока памяти и смещение в блоке. Биты смещения в адресе определяют часть блока памяти, запрашиваемую процессором. Адрес блока памяти, в свою очередь, разделяется на индекс и тэг.

Индекс используется для определения секции кэш-памяти, в которой может быть размещен данный блок. В кэш-памяти для каждого блока секции определяется тэг, представляющий собой старшую часть физического адреса. Биты тэгов хранятся в отдельной памяти. Посредством сравнения тэгов для блоков данной секции с входным физическим адресом определяется наличие запрашиваемого блока в кэш-памяти. Как правило, кэш-памяти второго уровня современных микропроцессоров делятся на несколько банков памяти для организации параллельного доступа. Существует два уровня деления на банки:

- Массив тэгов поделен на несколько банков (обычно на 4) для возможности одновременного доступа.

- Каждый банк тэгов поделен на несколько банков данных для возможности потокового доступа к банкам данных.

При определении наличия данного блока в кэш-памяти сравнению подвергаются только тэги, так как:

- Смещение не должно быть использовано, поскольку проверяется наличие блока памяти целиком.

- Проверка индекса является излишней, так как он используется при определении секции, которую необходимо проверить.

При одинаковом объеме кэш-памяти повышение ассоциативности приводит к возрастанию количества блоков, приходящихся на одну секцию. Следовательно, уменьшается размер индекса и увеличивается размер тэга.

Наиболее объективным показателем эффективности работы кэш-памяти является среднее время доступа, которое определяется формулой:

$$t_{\text{ср}} = t_{\text{попадание}} + n_{\text{промах}} \times t_{\text{промах}}, \quad (3)$$

где  $t_{\text{попадание}}$  – время обработки запроса, данные для которого присутствуют в кэш-памяти;  $n_{\text{промах}}$  –

отношение количества запросов, данные для которых отсутствуют в кэш-памяти, к общему количеству запросов к кэш-памяти;  $t_{\text{промах}}$  – время обработки запроса, данные для которого отсутствуют в кэш-памяти.

Из формулы (3) следует, что для увеличения производительности кэш-памяти необходимо уменьшить один или несколько параметров: время попадания, долю промахов и задержку при промахе. В данной статье рассматривается методика увеличения производительности кэш-памяти путём уменьшения времени попадания с помощью введения блоков буферизации.

#### V. КОНВЕЙЕР ОБРАБОТКИ КОМАНД СОХРАНЕНИЯ

С увеличением тактовой частоты микропроцессора увеличиваются относительные задержки при доступе к кэш-памяти. Для уменьшения задержек и увеличения пропускной способности кэш-памяти применяется конвейеризация доступа. Основными компонентами, вносящими задержки при обращении к кэш-памяти, являются считывание данных и тэгов, сравнение тэгов, запись или выдача данных, а также (при использовании помехоустойчивого кодирования) вычисление проверочных битов помехоустойчивого кода.

Конвейер обработки команды сохранения (рис. 1) в случае попадания в кэш-память второго уровня при отсутствии помехоустойчивого кодирования (рис. 1а) состоит из трёх стадий: поступление запроса, сравнение тэгов и запись данных. При включении помехоустойчивого кодирования (рис. 1б) конвейер дополняется стадией записи проверочных битов кода Хсяо. Каждая стадия занимает один такт процессорного времени.

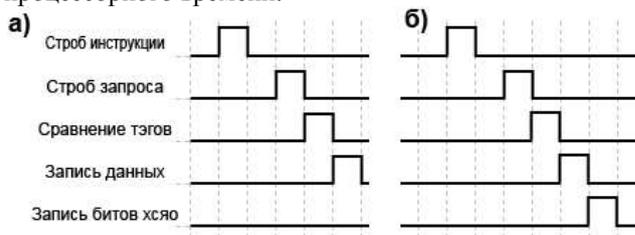


Рис. 1. Временные диаграммы, отображающие стадии конвейера обработки запроса на сохранение данных в кэш-памяти второго уровня: а) в отсутствие помехоустойчивого кодирования, б) помехоустойчивое кодирование включено

Данные для запроса на сохранение поступают на первой стадии конвейера и должны оставаться на шине до момента записи в кэш-память, т.е. до конца третьей стадии. Отсюда следует ограничение на частоту следования инструкций сохранения - интервал должен составлять не менее двух тактов между двумя инструкциями (рис. 2). Из временной диаграммы на рисунке 2 видно, что в данном случае на обработку двух запросов на сохранение необходимо 8 тактов процессорного времени без помехоустойчивого

кодирования (рис. 2а) и 10 тактов с помехоустойчивым кодированием (рис. 2б). Таким образом, при отсутствии блоков буферизации в кэш-памяти второго уровня для случая попадания в КД1 и К2 и отсутствия помехоустойчивого кодирования CPI = 3. Под CPI понимается количество тактов между двумя стробами инструкций в случае их последовательного поступления. При включении помехоустойчивого кодирования CPI возрастает до 4, так как в вычислении проверочных битов кода Хсяо задействовано большое количество логических элементов, что создаёт значительные временные задержки и не позволяет записать проверочные биты на такте записи данных. Также возникает необходимость в считывании данных из кэш-памяти второго уровня для проверки их с помощью помехоустойчивого кода и для перекодирования проверочных битов кода Хсяо.

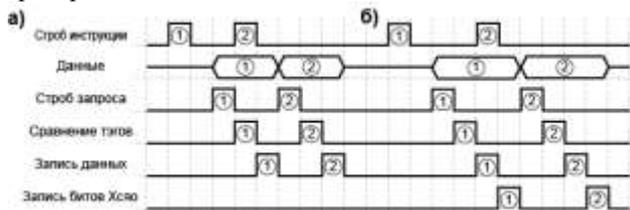


Рис. 2. Временная диаграмма, показывающая конвейер обработки запросов на сохранение в отсутствие входного буфера: а) в отсутствие помехоустойчивого кодирования, б) помехоустойчивое кодирование включено

## VI. БУФЕР ВХОДНЫХ ДАННЫХ

Для снятия ограничения на частоту следования инструкций сохранения, обусловленного необходимостью держать входные данные на стадии записи в кэш-память, предлагается введение двух регистров для хранения входных данных и двух регистров для хранения закодированного кода операции, логически объединённых в один блок. Данный блок назван буфером входных данных (БВД). Запись в БВД происходит на первой стадии конвейера.

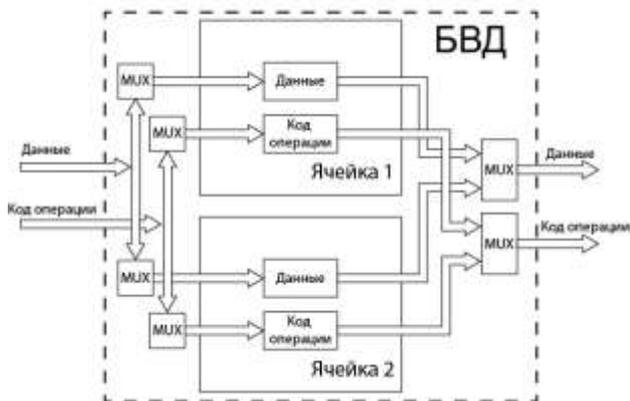


Рис. 3. Структура буфера входных данных

На рисунках 4 и 5 показаны временные диаграммы для кэш-памяти второго уровня с БВД при включенном и выключенном помехоустойчивом кодировании. Так как память данных и проверочных битов Хсяо в К2 разделены, запись данных и битов Хсяо производится от двух разных запросов на одном такте (рис. 5). Из рисунка 4 видно, что CPI = 1. Таким образом, для кэш-памяти второго уровня при отсутствии помехоустойчивого кодирования введение БВД позволяет сократить задержку до минимума.

Для снижения площади кристалла память данных К2, как правило, проектируется однопортовой. Из-за необходимости считывания данных для проверки и пересчёта проверочных битов кэш-памятей, использующих однопортовую память данных, запись на стадии поступления запроса невозможна. Это обстоятельство накладывает дополнительное ограничение на частоту следования инструкций загрузки для кэш-памятей с помехоустойчивым кодированием.



Рис. 4. Временная диаграмма, отображающая обработку запросов на сохранение данных при наличии БВД в кэш-памяти второго уровня (помехоустойчивое кодирование отключено)

При включении помехоустойчивого кодирования возникает необходимость в считывании из К2 данных и проверочных битов, поэтому в данном случае CPI увеличивается до 2,5.

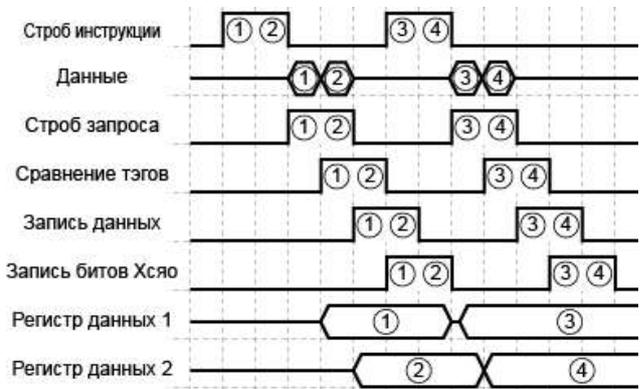


Рис. 5. Временная диаграмма, отображающая обработку запросов на сохранение данных при наличии БВД в кэш-памяти второго уровня (помехоустойчивое кодирование включено)

## VII. БУФЕР СОХРАНЕНИЯ ДАННЫХ

Из-за смещения записи данных и проверочных битов кода Хсяо частота следования инструкций уменьшается (см. рис. 5). Снизить задержки на запись данных от команд сохранения при попадании в кэш-память второго уровня при включённом помехоустойчивом кодировании позволяет введение дополнительного блока буферизации, называемого буфером сохранения данных (БСД). Данный блок состоит из двух ячеек (рис. 6), которые представляют собой совокупность данных для записи, адреса и проверочных битов кода Хсяо [7].

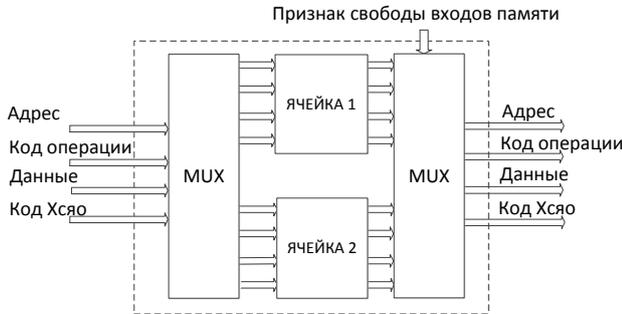


Рис. 6. Структура буфера сохранения данных

При введении БСД претерпевает изменения конвейер обработки данных в К2. Стадии записи данных и записи проверочных битов заменяются на стадию записи в БСД. Сама запись данных и проверочных битов перестает быть привязанной к какой-либо стадии конвейера. Запись происходит при освобождении входов памяти. На рисунке 5 показана обработка запроса на сохранение данных при введении в кэш-память второго уровня буфера сохранения данных. Из рисунка видно, что для шести инструкций CPI равен 2.

## VIII. ОЦЕНКА УСКОРЕНИЯ ПРИ ВВЕДЕНИИ БЛОКОВ БУФЕРИЗАЦИИ

Приняв среднюю частоту следования инструкций I за 0,3, можно оценить максимальное ускорение системы при введении блоков буферизации в кэш-памяти второго уровня.  $CPI_{\text{без хсяо}}$  – среднее количество тактов на инструкцию сохранения, попавшую в оба уровня кэш-памяти, при отключенном помехоустойчивом кодировании. Из рисунка 2а следует, что начальное  $CPI_{\text{без хсяо}}$  (т.е. в отсутствии блоков буферизации) равно 3.  $CPI_{\text{хсяо}}$  – среднее количество тактов на инструкцию сохранения, попавшую в оба уровня кэш-памяти, при наличии помехоустойчивого кодирования.

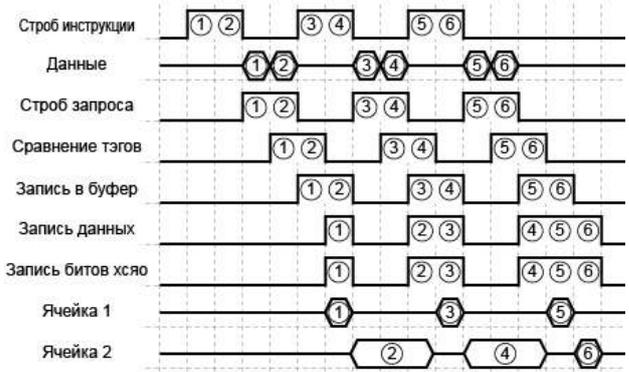


Рис. 7. Временная диаграмма, отображающая обработку запросов на сохранение данных при наличии БВД и БСД

Из рисунка 2б видно, что начальное  $CPI_{\text{хсяо}}$  (т.е. в отсутствии блоков буферизации) равно 4. С помощью формулы для общего ускорения системы (1) можно оценить максимальное ускорение системы. Оценка проводится для двух начальных значений  $CPI = 2$  и  $CPI = 4$ .

Ускорение при введении БВД для кэш-памяти второго уровня без помехоустойчивого кодирования:

$$\text{Ускорение} = \frac{1}{1 - 0,3 \times \frac{3-1}{2}} = 1,43 \quad (CPI_{\text{общ}} = 2).$$

$$\text{Ускорение} = \frac{1}{1 - 0,3 \times \frac{4-1}{4}} = 1,18 \quad (CPI_{\text{общ}} = 4).$$

Ускорение при введении БВД для кэш-памяти второго уровня с помехоустойчивым кодированием:

$$\text{Ускорение} = \frac{1}{1 - 0,3 \times \frac{4-2,5}{2}} = 1,29 \quad (CPI_{\text{общ}} = 2),$$

$$\text{Ускорение} = \frac{1}{1 - 0,3 \times \frac{4-2,5}{4}} = 1,13 \quad (CPI_{\text{общ}} = 4).$$

Ускорение при введении БВД и БСД для кэш-памяти второго уровня с помехоустойчивым кодированием:

$$\text{Ускорение} = \frac{1}{1 - 0,3 \times \frac{4-2}{2}} = 1,43 \quad (CPI_{\text{общ}} = 2),$$

$$\text{Ускорение} = \frac{1}{1 - 0,3 \times \frac{4-2}{4}} = 1,18 \quad (CPI_{\text{общ}} = 4).$$

Таким образом, теоретическое максимальное ускорение системы без помехоустойчивого кодирования при введении БВД совпадает с ускорением при введении двух блоков буферизации для кэш-памяти с помехоустойчивым кодированием.

При количественной оценке не учитывались размеры кэш-памятей К2 и КД1, так как в этом случае значения ускорения будут сильно зависеть от задержек на передачу данных из К2 в КД1 (после полного заполнения КД1) и от задержек при обратной записи (после полного заполнения К2), и, следовательно, значение ускорения будет почти полностью определяться именно этими параметрами. Таким образом, полученные при оценке значения можно считать максимально возможными.

## IX. ТЕСТИРОВАНИЕ И ОЦЕНКА РЕЗУЛЬТАТА

Определение значения ускорения системы при введении блоков буферизации проводилось на модели суперскалярного высокопроизводительного процессора. Объем К2 данного процессора составляет 256 КБ, объем КД1 – 16 КБ. Обе кэш-памяти имеют наборно-ассоциативную организацию с количеством наборов равным четырём. В качестве метода вытеснения данных в кэш-памяти используется политика LRU (Last Recently Used). Для замещения данных в кэш-памяти второго уровня используется обратная запись, в кэш-памяти данных первого уровня – сквозная запись.

Тестирование проводилось с помощью простейшей операции копирования с изменением размера копируемых массивов (параметр N) в пределах от 32 до 32768.

Результаты тестирования показаны на рисунке 8. Среднее ускорение составляет 17,1 %. Максимальное ускорение достигает 22% при размере массива N=2048.

За начальные значения CPI были приняты значения, полученные при моделировании кэш-памяти с помехоустойчивым кодированием при отсутствие блоков буферизации. Из рис. 8 видно, что CPI для размеров массивов до N=2048 находится в диапазоне от 3,5 до 4. Для N больше 2048 CPI приблизительно равен 2. Согласно расчётам по предлагаемой методике, произведённым для значений CPI = 2 и CPI = 4, теоретическое ускорение составляет 43 % и 18 %, соответственно. Практические значения ускорения составляют 17,1 % (N ≤ 2048 или при CPI ≈ 4) и 17,2 % (N > 2048 или при CPI ≈ 2). Отличие теоретического ускорения от полученного при CPI ≈ 2 обусловлено несколькими причинами:

- значительным увеличением количества операций обратной записи в память при N больше 2048 вследствие почти полного заполнения кэш-памяти второго уровня;
- невозможностью непрерывного запуска инструкций сохранения;
- низкой эффективностью развертывания циклов.

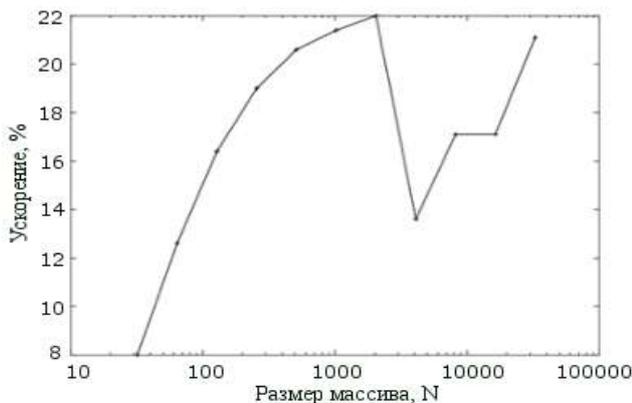


Рис. 8. Ускорение при введении БВД и БСД для К2 с помехоустойчивым кодированием (в процентах)

На рисунке 8 видно резкое уменьшение ускорения для N, находящихся в диапазоне от 2048 до 8096. Оно объясняется полным заполнением кэш-памяти первого уровня и связанными с ним дополнительными задержками при транспортировке данных из К2 в КД1. Увеличение значения ускорения при N > 8096 обусловлено уменьшением доли запросов, не попавших в К2 и КД1 одновременно (N < 2048).

## X. ВЫВОДЫ

Предложенная методика оценки эффективности кэш-памяти второго уровня позволяет на этапе создания её RTL-модели оптимизировать быстродействие микропроцессора при незначительном увеличении числа транзисторов. Введение блоков буферизации позволило в среднем увеличить скорость обработки до 17 %. Для дальнейшего увеличения эффективности работы кэш-памяти необходимо наряду с применением блоков буферизации введение дополнительных блоков: блока буферизации данных для обратной записи, блока буферизации для некэшируемых запросов и блока предварительной выборки данных. Развитие блока сохранения данных можно осуществить путем увеличения числа ячеек, а также путем внедрения функции слияния данных от запросов, следующих в одну строку кэш-памяти (так называемая «склейка» данных). С помощью данных решений возможно поднять частоту следования инструкций сохранения до 1 инструкции за такт.

Работа выполнена в рамках проекта РФФИ № 13-07-12062 «Фундаментальные проблемы создания микропроцессоров и коммуникационных сред супер-ЭВМ экзафлопсного класса, ориентированных на предсказательное моделирование задачи горения».

## ЛИТЕРАТУРА

- [1] URL: <http://cs.wisc.edu/multifacet/misc/spec2000cache-data> (дата обращения: 05.12.2013).
- [2] Jouppi N. Cache Write Policies and Performance // IEEE ACM SIGARCH Computer Architecture News. 1993. V. 21. № 2. P. 191–201.
- [3] John L. Hennessy, David A. Patterson Computer Architecture: A Quantitative Approach. Morgan Kaufmann Publishers. 2006. P. 40-42, 381-382.
- [4] Amdahl, Gene Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities // AFIPS Conference Proceedings. 1967. P. 483-485.
- [5] Jarek Nieplocha, Robert Harrison, Ian Foster Explicit Management of Memory Hierarchy. 1996. P. 3.
- [6] Sean White High Performance Power Efficient x86-64 Server And Desktop Processors Using the core code named “Bulldozer”. 2011.
- [7] Аряшев С.И., Корниленко А.В., Зубковская Н.В., Саяпин П.В. Повышение производительности подсистемы памяти методом буферизации данных // Информационные технологии. 2013. № 6. С. 11–17.