

УДК 004.25

DOI: 10.15827/0236-235X.116.078-084

Дата подачи статьи: 16.03.16

2016. Т. 29. № 4. С. 78–84

ИЗМЕРЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ КОМПОНЕНТОВ ПОДСИСТЕМЫ ПАМЯТИ ДЛЯ ГЕТЕРОГЕННЫХ СИСТЕМ НА КРИСТАЛЛЕ

А.О. Ключев, к.т.н., доцент, kluchev@cs.ifmo.ru;

А.А. Антонов, аспирант, 153287@niuitmo.ru

*(Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики (Университет ИТМО),
Кронверкский просп., 49, г. Санкт-Петербург, 197101, Россия)*

Проектирование специализированных вычислительных систем на базе гетерогенных платформ в рамках современных методологий предусматривает наличие модели системной архитектуры с информацией о характеристиках входящих в ее состав компонент. Производительность подсистемы памяти как ключевого связующего элемента в современных архитектурах является одной из важнейших характеристик, определяющих общесистемную производительность. Тенденция к интегрированию множества гетерогенных компонент в составе систем и сетей на кристалле, в том числе на уровне иерархии кэш-памяти, вносит проблемы при определении параметров реальных вычислительных платформ в силу того, что внутрисистемные тракты обмена оказываются технологически недоступными для прямых измерений, а общедоступная документация, как правило, лишь фрагментарно описывает внутреннюю организацию системы. Существующие методы непрямого измерения производительности компонент кэша не гарантируют соответствие получаемой модели кэш-памяти реальному поведению исследуемой системы.

В статье предложен метод непрямого селективного измерения производительности отдельных компонент кэш-подсистемы, в рамках которого предусмотрен ряд технических приемов для верификации селективности измерений с использованием информации о фактическом количестве обращений к отдельным компонентам кэш-подсистемы. Проанализирована применимость метода для нескольких популярных типов систем на кристалле и приведены результаты его апробации на процессоре Intel Core i7 и в заказной вычислительной платформе для проектирования гетерогенных измерительных систем реального времени.

Ключевые слова: *встроенные системы, системы на кристалле, гетерогенные системы, реальное время, кэш-память, системный уровень, согласованное проектирование аппаратного и программного обеспечения.*

Гетерогенные системы используются в различных областях для создания высокопроизводительных и энергоэффективных решений, таких как информационно-измерительные системы, аудио- и видеосистемы, системы управления, оборудование сетей и др. Под гетерогенными обычно понимают системы с аппаратным параллелизмом программно-видимых процессов или потоков и с различной степенью специализации вычислительных элементов: от процессоров общего и специального назначения до узкоспециализированных функциональных ядер: акселераторов вычислительных функций, контроллеров ввода/вывода и других, реализованных в виде сложнофункциональных блоков (СФ-блок или IP-ядро) на специализированной интегральной микросхеме (application specific integrated circuit, ASIC) или на ПЛИС [1]. Помимо набора вычислительных элементов, гетерогенность также проявляется в подсистемах коммуникации между ними: неоднородных сетевых структурах, а также распределенной памяти с многоуровневым кэшированием.

Моделирование интегрированных гетерогенных вычислительных платформ в проектировании

Сложность и многообразие вариантов аппаратно-программной организации таких систем, между которыми ведется поиск компромисса по их

характеристикам, вкупе с нарастающими требованиями к темпам и качеству проектирования ведут к необходимости развития и внедрения комплексных методологий системного уровня (hardware-software co-design, platform-based design, ESL-based co-design), в которых особое внимание уделяется этапам раннего архитектурного и микроархитектурного проектирования вычислительной системы без фиксации деталей аппаратно-программной реализации ее компонент и связей между ними [2]. Данные этапы предусматривают разработку функциональной модели системы («золотой модели», как правило, исполняемой) и параметрической модели вариантов системной архитектуры для оценки нефункциональных характеристик системы. Далее, в рамках так называемого исследования проектного пространства (design space exploration), производятся итеративное аппаратно-программное разделение и отображение «золотой модели» на различные варианты архитектур и оценка полученных результатов. Метрики, применяемые в модели вариантов системной архитектуры, характеризуют элементы используемой базовой аппаратно-программной платформы (СФ-блоки, коммуникации, комплексные механизмы) и включают в себя производительность процессоров, энергопотребление, емкость памяти, пропускную способность каналов связи и др. [1]. Пример инструмента, реализующего вышеупомянутый подход, – SystemCo-Designer [3].

Растущие требования как к разнообразию функций, реализуемых вычислительными системами на сегодняшний день, так и к их нефункциональным характеристикам обуславливают тенденцию к применению вычислительных платформ на базе единой СБИС (технологий «система на кристалле» (СнК) и «сеть на кристалле» [4]), в рамках которой интегрировано множество компонентов повторного использования (процессоров общего назначения, видеопроцессоров, DSP и прочих блоков), называемых IP-ядрами, на этапе проектирования. Количество IP-ядер, интегрируемых в рамках СнК, неуклонно растет, что вкупе с фундаментальными ограничениями по целостности и задержке распространения сигналов ведет к усложнению организации коммуникационной подсистемы и, соответственно, к усложнению ее проектирования и оценке ее реальных характеристик [5]. С учетом того, что на сегодняшний день архитектурно обмен через общую память доминирует в роли коммуникации между процессорными элементами, измерение производительности внутрисистемных трактов обмена с подсистемой памяти становится неотъемлемой частью моделирования вычислительных платформ в составе маршрутов проектирования на их основе.

Проблемы измерения производительности подсистемы памяти в интегрированных вычислительных платформах

Существующая в современных архитектурах тенденция к интегрированию множества различных специализированных компонент в рамках единой СнК усугубляет проблемы как технологического, так и методологического плана. Блоки в составе СнК зачастую не только физически реализуются на одном кристалле, но и интегрируются на уровне иерархии памяти, в результате чего появляется возможность ускоренного обмена данными между ними. Интегрированные подсистемы памяти для гетерогенных архитектур на сегодняшний день являются актуальной темой исследований (например, кэш-когерентные подсистемы в [6]) и интенсивно внедряются в промышленных изделиях. В качестве примера можно назвать технологию так называемой гетерогенной системной архитектуры (heterogeneous systems architecture, HSA) для СнК [7], развиваемую компаниями AMD, ARM, Imagination, Qualcomm и др., в которой декларируется возможность унифицированного доступа к памяти со стороны различных компонент в составе гетерогенной архитектуры. Примерами реальных СнК, в составе которых на уровне иерархии памяти интегрированы процессорные элементы различной степени специализации, являются СнК Intel Core i7 4XXX (Haswell) и 5XXX (Broadwell), AMD Carrizo, Xilinx Zynq-7000 и UltraScale, TI OMAP и DaVinci. В качестве примера

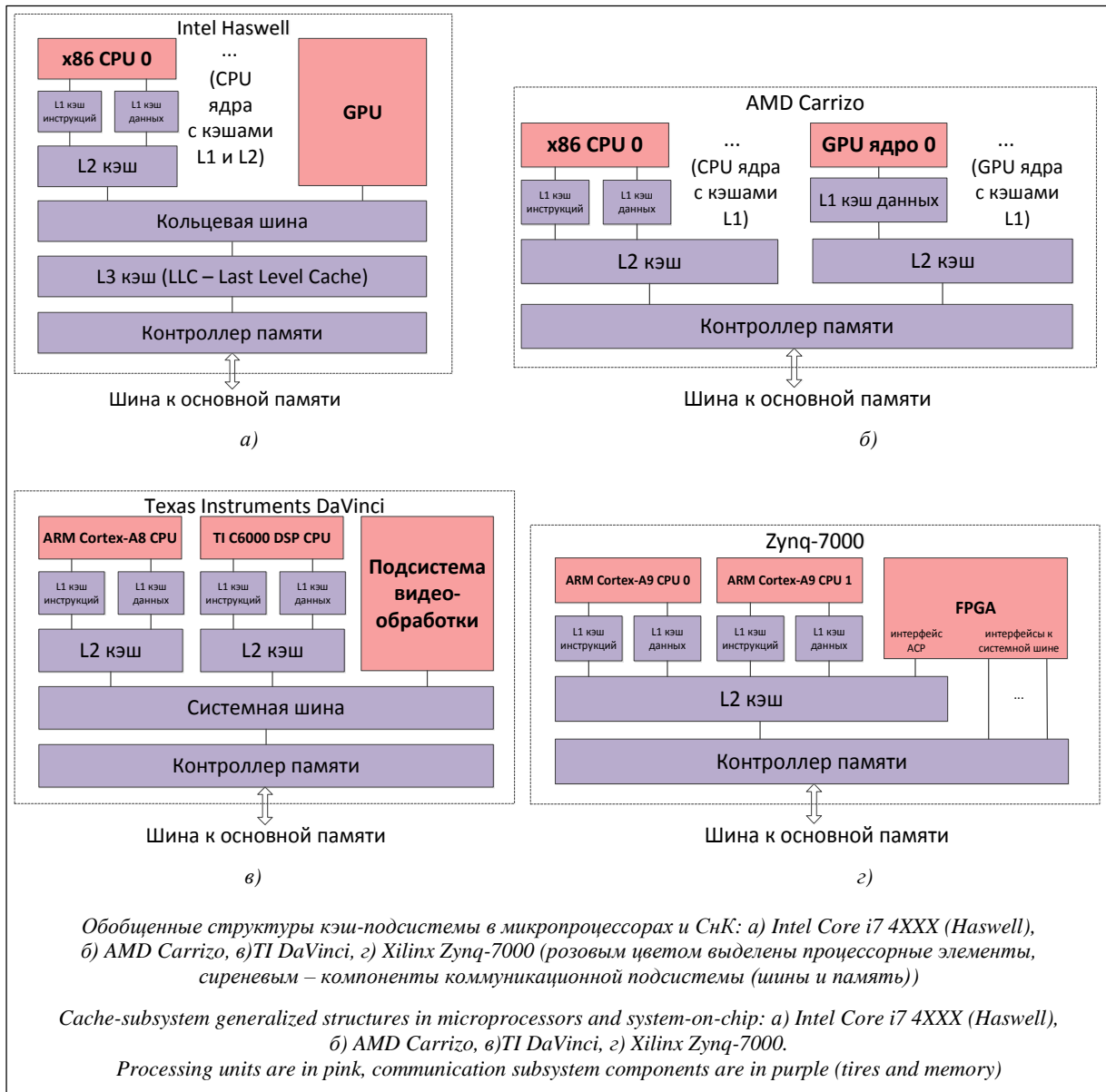
на рисунке приведены обобщенные структуры кэш-подсистем для некоторых из упомянутых платформ.

С точки зрения использования вышеупомянутых платформ в проектировании, внутрикристалльная реализация коммуникаций между блоками в составе гетерогенной архитектуры обуславливает фактическое отсутствие физического доступа к ним и способствует тому, что системный проектировщик крайне ограниченно осведомлен о деталях организации этих коммуникаций. Таким образом, необходимость работы с платформой на уровне отдельных вычислительных компонент и связей между ними вступает в противоречие с тем, что структура кэш-подсистемы, топология коммуникационной сети, протоколы обмена и так далее, как правило, относятся к микроархитектуре системы, а детали микроархитектурной реализации реальных коммерческих СБИС обычно являются коммерческой тайной соответствующих фирм-производителей и крайне поверхностно или фрагментарно описаны в документации, находящейся в открытом доступе. Приводимые оценки обычно являются оптимистичными и редко наблюдаются в реальности. Задача моделирования интегрированных вычислительных платформ для последующего отображения функциональной модели системы или оценки (верификации) показателей для разработанных систем на реальных образцах делает актуальным развитие методов и технологий селективного непрямого измерения производительности внутрисистемных трактов обмена.

Пример такого исследования – работа [8], где с использованием СнК Xilinx Zynq-7000, имеющей интерфейс для подключения к иерархии кэш-памяти со стороны блока программируемой логики, измерялась производительность данного интерфейса на примере задачи КИХ-фильтрации образа сигнала. Однако, к сожалению, большинство подобных исследований так или иначе привязаны к какой-либо платформе или приложению, и проблема характеристики внутрисистемных каналов связи в целом и особенно иерархии кэш-памяти остается на данный момент открытой.

Существующие методы и технологии не прямых измерений на уровне компонентов кэш-подсистем

В то время как оценке производительности вычислительных систем на примере тех или иных приложений посвящено значительное количество работ отечественных и зарубежных авторов, «погружение» в детали микроархитектуры реальных систем, организация измерений применительно к отдельным подсистемам и трактам обмена, что необходимо для планирования вычислений в рамках гетерогенных архитектур, является существенно более сложной и изощренной задачей.



К тому же эта область становится все более нетривиальной в силу развития и, соответственно, усложнения организации исследуемых систем. Однако можно назвать определенное количество разработок в данной области.

Большинство алгоритмов анализа характеристик компонентов иерархии кэш-памяти основано на наборе классических приемов, которые можно найти, например, в [9]. Измерение заключается в оценке времени выполнения специальной процедуры, осуществляющей обращения к памяти по определенным шаблонам. Рабочей процедурой является циклическое обращение к элементам массива, размещенного в памяти. Размер массива варьируется от минимального до максимального размера кэша (обычно удвоением). Обращение производится с прореживанием, которое позволяет обеспечить промахи по кэшу и также варьируется от нуля до максимального размера кэша. Повтор-

ные обращения к тем же адресам позволяют приблизительно оценить накладные расходы на измерения. В то же время в части интерпретации результатов и реальной микроархитектурной селективности измерений есть ряд проблем, в число которых входит необходимость вычисления размеров массива и адресов, возможность аппаратной параллелизации запросов, сложность последовательного размещения в силу работы механизма виртуальной памяти и так далее.

В рамках исследований в области адаптивного ПО в Корнелльском университете был разработан фреймворк X-Ray [10–12]. Он содержит набор измерительных средств (бенчмарков) микроархитектурного уровня (микробенчмарков) для оценки различных аспектов системной производительности. Посредством варьирования размера рабочего массива и степени прореживания производится измерение размера кэша, размера строки кэша, ассоци-

ативности и задержки. Последовательное размещение рабочего массива в памяти достигается через запрос больших страниц, специфический для операционной системы. Однако соответствие используемой модели реальной системе (то есть тех механизмов, параметры которых определяются, и уже реализованных в системе) остается проблемой.

Фреймворк LMBench [13] также предоставляет множество инструментов для измерения системной производительности с различных точек зрения – от производительности системных вызовов ОС до измерений микроархитектурного уровня. Например, имеется микробенчмарк, позволяющий произвести измерения размера кэша, размера строки кэша, задержки и ассоциативности для одного уровня кэша. Компания Intel рекомендует использовать LMBench для исследования иерархии памяти [14]. На графике зависимости пропускной способности и задержки от размера рабочего массива можно увидеть «ступеньки», которые соответствуют определенным уровням иерархии памяти, и таким образом оценить целевые характеристики.

Исследовательская группа Технического университета Дрездена проводила измерения производительности компонентов иерархии памяти для микропроцессоров Intel Nehalem, Sandy Bridge и AMD Bulldozer [15, 16]. Измерения выполнялись в пакете BenchIT. В системе инициализировались несколько потоков, которые отображались на различные процессорные ядра. Посредством определенной последовательности запросов кэш-память вводилась в начальное состояние. Далее с помощью встроенного в устройство таймера была измерена задержка кэш-памяти различных уровней и основной памяти. «Потолок» пропускной способности достигался посредством запуска нескольких потоков, которые обращались к одному и тому же компоненту памяти.

Более точный подход был предложен в [17]. Метод ориентирован на измерение кратковременных событий: например, увеличение задержки при наполнении регистров MSHR (miss information/status handling registers). Поскольку такие события нельзя измерить, усредняя результаты длительных измерений, авторы предлагают так называемую методологию SETE, которая предусматривает, например, использование встроенного высокоточного таймера и «разогрев» кэша инструкции путем выполнения нескольких запусков эксперимента с учетом только последнего результата.

В целом в существующих методах и технологиях измерений производительности компонентов кэш-подсистемы можно выделить ряд открытых проблем, связанных с возможным искажением результатов измерений. При подходе с постепенным увеличением размера рабочего массива и анализа «ступенек» на графике времени выполнения теста по факту обращения производятся к различным

уровням кэш-памяти, что затрудняет выделение случаев обращения именно к интересующему уровню кэш-памяти. Если структура теста ориентирована на определенный вариант организации кэш-памяти, без достоверной информации о том, что в конкретной системе реализован именно этот вариант, реальная селективность измерений также остается под вопросом, что допускает неверную интерпретацию результатов измерений.

Метод селективного измерения производительности компонентов кэш-памяти

Для решения проблемы обеспечения реальной селективности измерений авторами был разработан оригинальный метод селективного измерения производительности компонентов кэш-памяти. В целом данный метод основан на работе [9] с использованием следующих технических приемов.

Тесты представляют собой зацикленные обращения к памяти без вычислений между итерациями. Доступ к конкретному уровню кэша достигается посредством параметризации размера рабочего массива (он должен быть меньше, чем размер целевого кэша, но больше, чем кэш меньшего уровня). При первичной инициализации массива данные размещаются в кэше нужного уровня, в то время как последующие обращения формируют рабочую процедуру.

Требования к значениям элементов массива различаются в зависимости от измеряемой характеристики. Тесты пропускной способности не предъявляют каких-либо специфических требований. Для измерения задержки требуется, чтобы каждый элемент содержал адрес последующего элемента, что позволяет убрать влияние параллелизации и конвейеризации доступа к памяти. В литературе данный подход носит название «погоня за указателем» (pointer chasing) [15]. Для недопущения попадания в кэши более низкого уровня применяется прореживание доступа.

Однако ввиду сложности механизмов управления кэш-памятью на сегодняшний день гарантировать обращения к требуемому уровню кэша без дополнительных проверок не представляется возможным. Другими словами, должно подтверждаться соответствие используемой модели иерархии памяти и ее реальной организации в системе. Поэтому с помощью встроенных средств мониторинга (так называемых счетчиков производительности) осуществляется проверка соответствия реального поведения подсистемы памяти ожидаемому (то есть специфицированному в рамках принятой модели подсистемы памяти).

Работа тестовой процедуры сопровождается подсчетом обращений к компонентам подсистемы памяти с помощью встроенных средств мониторинга. Запуск и останов счетчиков жестко синхронизируются с основной рабочей процедурой.

Встроенные счетчики обращений к компонентам кэш-памяти в различных вычислительных платформах**Built-in counters of references to memory cache components in different computing platforms**

Вычислительная платформа	Количество уровней кэш-памяти	Наличие встроенного блока измерения производительности	Количество счетчиков производительности	Доступные события кэш-подсистемы, актуальные для применения метода
Intel Haswell [18]	3	+	3 (назначение фиксировано) + 8 (программируемых)	Количество попаданий и промахов по кэшам L1, L2, L3
AMD Carrizo [19]	2	+	6 (на ядро) + 4 (сверный мост)	Количество попаданий в кэши L1 и L2
Texas Instruments DaVinci [20]	2	+	4	Количество обращений и промахов по кэшам L1 и L2
Xilinx Zynq-7000 [21]	2	+	6	Количество обращений и промахов по кэшу L1 инструкций и данных

Варьирование параметров производится до тех пор, пока не будут достигнуты целевые значения. В качестве начальных параметров следует брать размер рабочего массива в $\frac{1}{2}$ от исследуемого компонента кэш-памяти и шаг в пределах строки кэша для кэш-памяти 1-го уровня и, соответственно, кратный размеру строки для кэш-памяти последующих уровней. Варьирование следует производить согласно следующим правилам:

- если результаты опыта показали, что попаданий в кэш нужного уровня меньше ожидаемых за счет попаданий в кэш более низкого уровня, значит, сработала загрузка данных в кэш более низкого уровня, и поэтому следует увеличить степень прореживания обращений; если прореживание обращений достигает размера кэша более низкого уровня, это, скорее всего, означает, что аппаратное обеспечение детектировало шаблон обращений к памяти, следовательно, величину прореживания надо изменить;

- если результаты опыта показали, что попаданий в кэш нужного уровня меньше ожидаемых за счет попаданий в кэш более высокого уровня, значит, часть рабочего массива была выгружена из кэша нужного уровня с целью освобождения места для новых потенциально нужных данных, поэтому следует уменьшить размер рабочего массива с целью минимизации вероятности выгрузки данных в составе его в кэш более высокого уровня.

В результате, итеративно повторяя эксперименты в соответствии с приведенными правилами и используя информацию встроенных блоков измерения производительности, можно определить выборочную пропускную способность и задержку отдельных компонентов кэш-подсистемы в составе иерархии памяти целевой вычислительной платформы. Использование предложенных приемов позволяет верифицировать действительное количество обращений к компонентам иерархии памяти и за счет итеративных повторов измерений добиться реальной селективности измерений.

Применимость метода селективного измерения производительности компонентов кэш-памяти

Требования к наличию средств встроенного мониторинга производительности кэш-подсистемы является фактором, ограничивающим применимость представленного метода только теми системами, где данные средства реализованы. Исходя из содержания метода, для его применимости требуются как минимум счетчик циклов (таймер) и счетчик, программируемый на успешные обращения на каждый из интересующих компонентов кэш-подсистемы. Данные о доступности встроенных средств мониторинга производительности для рассмотренных ранее вычислительных платформ приведены в таблице.

Несмотря на то, что количество счетчиков и наличие тех или иных событий являются специфическими для платформы, можно видеть, что большинство современных вычислительных платформ соответствуют данным требованиям. Исключением в приведенных примерах является Xilinx Zynq-7000, для которого блок мониторинга производительности предусматривает подсчет только тех 58 событий, которые относятся непосредственно к ядру Cortex-A9 [21]. Поскольку кэш L2 для Cortex-A9 является внешним по отношению к ядру, отличить попадание в L2 от промаха, видимо, возможно только для подконтрольных трактов данных (например, при обращении к области памяти, отображенной на встроенную FPGA).

В рамках апробации метода был разработан пакет программных тестов для микропроцессора Intel Core i7 4770K [22]. Тесты были реализованы в виде модуля для загрузчика GNU GRUB 2.0 в среде без операционной системы, что нивелировало влияние ее эффектов на измерения. С помощью предложенного метода были проведены измерения с достижением соответствия общего количества запросов к

памяти и действительного количества обращений к каждому из исследуемых уровней кэш-подсистемы и коэффициентом вариации в пределах 10 %.

Также авторы использовали описанный метод в реальном проектировании на базе разработанной авторами заказной гетерогенной платформы для систем цифровой обработки сигналов [23]. В рамках данной платформы была реализована система обработки радиосигналов реального времени на процессорах ЦОС, ПЛИС и специализированных интегральных микросхемах. Предложенный метод позволил верифицировать селективную производительность внутрисистемных трактов обмена и осуществить прогнозирование предельной длительности циклов, необходимой для обработки отсчетов входного сигнала.

Таким образом, в соответствии с перспективными методологиями проектирования специализированных гетерогенных вычислительных систем необходимость моделирования системной платформы делает актуальной задачу селективного измерения пропускной способности и задержки коммуникационных компонентов между процессорными элементами в составе платформы. Тенденция к интеграции разнородных процессоров в рамках единой СБИС и, соответственно, единой иерархии памяти и внутрикристалльных коммуникаций вносит проблемы при осуществлении таких измерений. Для решения данной проблемы в статье предлагается оригинальный метод селективного измерения производительности компонентов кэш-памяти. Метод позволяет производить измерения пропускной способности и задержки отдельных уровней кэш-памяти с контролем селективности измерений с помощью данных о фактическом количестве обращений к отдельным блокам иерархии памяти. Предложенный метод был апробирован в рамках исследования кэш-подсистем микропроцессора Intel Core i7 и специализированной гетерогенной вычислительной платформы реального времени. Разработанный пакет тестов может использоваться для проведения измерений производительности компонентов кэш-подсистем для актуальных гетерогенных вычислительных платформ.

Литература

1. Pomante L., Serri P., Incerto E., Volpe J. HW/SW co-design of heterogeneous multiprocessor dedicated systems: a systemc-based environment. Proc. 2nd World Congr. Multimed. Comput. Sci. 2014, pp. 9–11.
2. Teich J. Hardware/software codesign: the past, the present, and predicting the future. Proc. IEEE. 2012, vol. 100, pp. 1411–1430.
3. Keinert J., Streubühr M., Schlichter T., Falk J., Gladigau J., Haubelt C., Teich J., and Meredith M. SystemCoDesigner – an automatic ESL synthesis approach by design space exploration and behavioral synthesis for streaming applications. ACM Trans. Des. Autom. Electron. Syst. 2009, vol. 14, no. 1, article 1, pp. 1–23.
4. Исаев М.В., Кожин А.С., Костенко В.О., Поляков Н.Ю., Сахин Ю.Х. Двухъядерная гетерогенная система на кристалле

«Эльбрус-2С+» // Вопросы радиоэлектроники. 2012. Т. 4. № 3. С. 42–52.

5. Ahmad B. Communication centric platforms for future high data intensive applications. PhD Thesis, Univ. of Edinburgh, 2009, 149 p.
6. Hechtman B.A., Sorin D.J. Evaluating cache coherent shared virtual memory for heterogeneous multicore chips. Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw., 2013, pp. 118–119.
7. HSA Foundation. URL: <http://www.hsafoundation.com/> (дата обращения: 15.03.2016).
8. Sadri M., Weis C., Wehn N., and Benini L. Energy and Performance Exploration of Accelerator Coherency Port Using Xilinx ZYNQ. Proc. 10th FPGAWorld Conf. FPGAWorld'13. 2013, no. 5, 5 p. URL: http://www.googoolia.com/downloads/papers/sadri_fpgaworld_ver2.pdf (дата обращения: 15.03.2016).
9. Hennessy L.J., Patterson A.D. Computer architecture: a quantitative approach. Morgan Kaufmann Publ., 2003, 1141 p.
10. Yotov K. On the role of search in generating high performance BLAS libraries. PhD Thesis, Cornell Univ., 2006, 194 p.
11. Yotov K., Pingali K., Stodghill P. Automatic measurement of memory hierarchy parameters. Proc. 2005 ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst. SIGMETRICS'05, NY, USA, ACM Press, 2005, 181 p.
12. Yotov K., Pingali K., Stodghill P. Automatic measurement of hardware parameters for embedded processors. 2005. URL: <http://ecommons.library.cornell.edu/bitstream/1813/5674/1/TR2005-1974.pdf> (дата обращения: 15.03.2016).
13. LMBench – Tools for Performance Analysis. URL: <http://www.bitmover.com/lmbench/lmbench.html> (дата обращения: 15.03.2016).
14. Measuring cache and memory latency and cpu to memory bandwidth for use with intel architecture. White Paper (Intel). 2008, pp. 1–14.
15. Molka D., Hackenberg D., Schone R., and Muller M.S. Memory performance and cache coherency effects on an intel nehalem multiprocessor system. Proc. 18th Int. Conf. Parallel Archit. Compil. Tech. IEEE, 2009, pp. 261–270.
16. Molka D., Sch R. Main memory and cache performance of intel sandy bridge and AMD Bulldozer Memory size. MSPC'2014. URL: https://tu-dresden.de/die_tu_dresden/zentrale_einrichtungen/zih/forschung/projekte/benchit2014_MSPC_authors_version.pdf (дата обращения: 15.03.2016).
17. Fang Z., Mehta S., Yew P.-C., Zhai A., Greensky J., Gautham B., Binyu Z. Measuring microarchitectural details of multi- and many-core memory systems through microbenchmarking. ACM TACO, 2014, vol. 11, no. 4, article 55, pp. 1–26.
18. Intel® 64 and IA-32 Architectures Software Developer's Manual. Combined volumes. June 2015. URL: <http://www.intel.de/content/www/de/de/architecture-and-technology/64-ia-32-architectures-software-developer-manual-325462.html> (дата обращения: 15.03.2016).
19. BIOS and Kernel Developer's Guide (BKDG) for AMD Family 15h Models 60h-6Fh Processors. AMD. June 2015. URL: http://support.amd.com/TechDocs/50742_15h_Models_60h-6Fh_BKDG.pdf (дата обращения: 15.03.2016).
20. CortexTM-A8 Technical Reference Manual. Revision: r3p2. ARM. URL: <https://static.docs.arm.com/ddi0344/k/DDI0344.pdf> (дата обращения: 15.03.2016).
21. CortexTM-A9 Technical Reference Manual. Revision: r2p2. ARM. URL: http://infocenter.arm.com/help/topic/com.arm.doc/ddi0409f/DDI0409f_cortex_a9_neon_mpe_r2p2_trm.pdf (дата обращения: 15.03.2016).
22. Антонов А.А. Организация измерений производительности компонентов подсистемы памяти микропроцессоров и СнК. Магистер. дис. СПб: Изд-во Ун-та ИТМО, 2014. С. 1–95.
23. Kustarev P., Antonov A., Pinkevich V., Yanalov R. Using selective memory performance evaluation for time-critical embedded systems design. Proc. 15th Int. Multidiscip. Sci. Geoconf. SGEM-2015. URL: <http://www.sgem.org/SGEMLIB/spip.php?article5637> (дата обращения: 15.03.2016).

MEMORY SUBSYSTEM COMPONENTS PERFORMANCE EVALUATION FOR HETEROGENEOUS SYSTEMS-ON-CHIP

A.O. Klyuchev¹, Ph.D. (Engineering), Associate Professor, kluchev@cs.ifmo.ru; A.A. Antonov¹, Postgraduate Student, 153287@niuitmo.ru

¹ The National Research University of Information Technologies, Mechanics and Optics, Kronverksky Ave. 49, St. Petersburg, 197101, Russian Federation

Abstract. Specialized computer system design based on heterogeneous platforms within modern methodologies assumes a system platform model that contains information about the characteristics of its individual components. A memory subsystem is considered as a key communication element of current architectures, so its performance evaluation is one of the most important characteristics that determine the overall system performance. The tendency of multiple heterogeneous components integration within the systems and networks on-chip (including the cache memory hierarchy level) introduces new problems for computational platforms modeling, since in-system datapaths that connect the computational elements with cache memory hierarchy components become technologically inaccessible for direct measurements, while the publicly available documentation typically poorly describes the system's internal organization. Existing methods of indirect evaluation of cache subsystem components' performance do not guarantee the compliance of cache memory model being implied with the real behavior of the system under exploration.

The article proposes the method of indirect selective performance evaluation of individual cache subsystem components. The method provides a set of techniques for verification of evaluation selectiveness using information about the actual amount of requests to the individual cache subsystem components. The paper explores feasibility of the method using several actual instances of heterogeneous computer systems and provides the results of its approbation for Intel Core i7 processor and custom computational platform for real-time heterogeneous measurement systems design.

Keywords: embedded systems, systems-on-chip, heterogeneous systems, real-time, cache memory, Electronic System Level (ESL), HW/SW Co-Design.

References

1. Pomante L., Serri P., Incerto E. HW/SW Co-Design of Heterogeneous Multiprocessor Dedicated Systems: a SystemC-based Environment. *Proc. 2nd World Congr. Multimed. Comput. Sci.* 2014, pp. 9–11.
2. Teich J. Hardware/Software Codesign: The Past, the Present, and Predicting the Future. *Proc. IEEE*. 2012, vol. 100, Special Centennial iss., pp. 1411–1430.
3. Keinert J., Streubühr M., Schlichter T., Falk J., Gladigau J., Haubelt C., Teich J., Meredith M. SystemCoDesigner – an automatic ESL synthesis approach by design space exploration and behavioral synthesis for streaming applications. *ACM Trans. Des. Autom. Electron. Syst.* 2009, vol. 14, no. 1, pp. 1–23.
4. Isaev M.V., Kozhin A.S., Kostenko V.O., Polyakov N.Yu., Sakhin Yu.Kh. Dual-Core Heterogeneous System-on-Chip “Elbrus-2S+”. *Problems of Advanced Micro- and Nanoelectronic Systems Development (MES-2012)*. IDPM RAS Publ., 2012, vol. 4, no. 3, pp. 42–52 (in Russ.).
5. Ahmad B. *Communication Centric Platforms for Future High Data Intensive Applications*. PhD Thesis, Univ. of Edinburgh Publ., 2009, 149 p.
6. Hechtman B.A., Sorin D.J. Evaluating Cache Coherent Shared Virtual Memory for Heterogeneous Multicore Chips. *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, 2013, pp. 181–119.
7. HSA Foundation. Available at: <http://www.hsafoundation.com/> (accessed March 15, 2016).
8. Sadri M., Weis C., Wehn N., Benini L. Energy and Performance Exploration of Accelerator Coherency Port Using Xilinx ZYNQ. *Proc. 10th FPGAWorld Conf. FPGAWorld '13*. 2013, no. 5, 5 p. Available at: http://www.googoolia.com/downloads/papers/sadri_fpga_world_ver2.pdf (accessed March 15, 2016).
9. Hennessy L.J., Patterson A.D. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publ., 2003, 1141 p.
10. Yotov K. *On the Role of Search in Generating High Performance BLAS Libraries*. PhD Thesis, Cornell Univ., 2006, 194 p.
11. Yotov K., Pingali K., Stodghill P. Automatic measurement of memory hierarchy parameters. *Proc. 2005 ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst. SIGMETRICS '05*. NY, USA, ACM Press, 2005, 181 p.
12. Yotov K., Pingali K., Stodghill P. *Automatic Measurement of Hardware Parameters for Embedded Processors*. 2005. Available at: <http://ecommons.library.cornell.edu/bitstream/1813/5674/1/TR2005-1974.pdf> (accessed March 15, 2016).
13. *LMbench – Tools for Performance Analysis*. Available at: <http://www.bitmover.com/lmbench/lmbench.html> (accessed March 15, 2016).
14. Measuring Cache and Memory Latency and CPU to Memory Bandwidth for Use with Intel Architecture. *White Paper (Intel)*. 2008, pp. 1–14.
15. Molka D., Hackenberg D., Schone R., Muller M.S. Memory Performance and Cache Coherency Effects on an Intel Nehalem Multiprocessor System. *Proc. 18th Int. Conf. Parallel Archit. Compil. Tech. IEEE*. 2009, pp. 261–270.
16. Molka D., Sch R., Bridge-ep I.X.E.-S. Main Memory and Cache Performance of Intel Sandy Bridge and AMD Bulldozer Memory size. *MSPC 2014*. Available at: https://tu-dresden.de/die_tu_dresden/zentrale_einrichtungen/zih/forschung/projekte/benchit/2014_MSPC_authors_version.pdf (accessed March 15, 2016).
17. Fang Z., Mehta S., Yew P.-C., Zhai A., Greensky J., Gautham B., Binyu Z. Measuring Microarchitectural Details of Multi- and Many-Core Memory Systems through Microbenchmarking. *ACM TACO*. 2014, vol. 11, no. 4, pp. 1–26.
18. Intel® 64 and IA-32 Architectures Software Developer's Manual – Combined Volumes. 2015. Available at: <http://www.intel.de/content/www/de/de/architecture-and-technology/64-ia-32-architectures-software-developer-manual-325462.html> (accessed March 15, 2016).
19. BIOS and Kernel Developer's Guide (BKDG) for AMD Family 15h Models 60h-6Fh Processors. *AMD*. 2015. Available at: http://support.amd.com/TechDocs/50742_15h_Models_60h-6Fh_BKDG.pdf (accessed March 15, 2016).
20. CortexTM-A8 Technical Reference Manual. Revision: r3p2. *ARM*. Available at: <https://static.docs.arm.com/ddi0344/k/DDI0344.pdf> (accessed March 15, 2016).
21. CortexTM-A9 Technical Reference Manual. Revision: r2p2. *ARM*. Available at: http://infocenter.arm.com/help/topic/com.arm.doc/ddi0409f/DDI0409F_cortex_a9_neon_mpe_r2p2_trm.pdf (accessed March 15, 2016).
22. Antonov A. *Organizatsiya izmereny proizvoditelnosti komponentov podsystemy pamyati mikroprotessorov i SnK*. [Organization of Memory Components Performance Evaluation for Microprocessors and SoCs]. Master's Thesis. ITMO Univ., 2014, pp. 1–95 (in Russ.).
23. Kustarev P., Antonov A., Pinkevich V., Yanalov R. Using selective memory performance evaluation for time-critical embedded systems design. *Proc. 15th Int. Multidiscip. Sci. Geoconf. SGEM 2015*. Available at: <http://www.sgem.org/SGEMLIB/spip.php?article5637> (accessed March 15, 2016).