

УДК 004.2

## РАЗРАБОТКА ПРОТОКОЛА МНОЖЕСТВЕННОГО ДОСТУПА ДЛЯ ПРОЦЕССОРОВ С МНОГОУРОВНЕВЫМ КЭШИРОВАНИЕМ

А.А. Антонов<sup>a</sup>, А.О. Ключев<sup>a</sup>, М.С. Комар<sup>b</sup>, П.В. Кустарев<sup>a</sup>,  
Е.А. Кучерявый<sup>c</sup>, Д.А. Молчанов<sup>c</sup>, В.И. Петров<sup>d</sup>, А.Е. Платунов<sup>a</sup>

<sup>a</sup> Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

<sup>b</sup> Ярославский государственный университет им. П.Г. Демидова, Ярославль, 150000, Российская Федерация

<sup>c</sup> Технологический университет Тампере, Тампере, 33720, Финляндия

<sup>d</sup> Санкт-Петербургский университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация

Адрес для переписки: [platunov@lmt.ifmo.ru](mailto:platunov@lmt.ifmo.ru)

### Информация о статье

Поступила в редакцию 17.03.15, принята к печати 27.03.15

doi:10.17586/2226-1494-2015-15-3-463-469

Язык статьи – русский

**Ссылка для цитирования:** Антонов А.А., Ключев А.О., Комар М.С., Кустарев П.В., Кучерявый Е.А., Молчанов Д.А., Петров В.И., Платунов А.Е. Разработка протокола множественного доступа для процессоров с многоуровневым кэшированием // Научно-технический вестник информационных технологий, механики и оптики. 2015. Т. 15. № 3. С. 463–469.

### Аннотация

Приведены результаты исследования архитектуры современных процессоров с иерархической организацией внутренней памяти. Выполнен анализ возможности внедрения беспроводной сети для организации взаимодействия между вычислительными ядрами и общим кэшем последнего (третьего) уровня. Для формирования требований к беспроводному каналу связи определяется набор функций обмена сообщениями в современных процессорах общего назначения на примере Intel Core i7 (Haswell). Разработана модель взаимодействия компонентов кэш-подсистемы процессора, на базе которой оцениваются характеристики обмена между ними. Анализ модели показывает, что задержка предлагаемого беспроводного канала при передаче пакета данных в кэш-подсистеме составит 0,26 нс, что хорошо соотносится с задержкой обращения к кэшу самого первого уровня (около 2 нс). Беспроводной канал удовлетворяет требованиям по дальности, предполагая возможность передачи данных на расстояние до 3 см, и энергоэффективности, расходуя мощность 1 мкВт. Результатом работы является протокол множественного доступа, предназначенный для реализации в рамках беспроводной сети на кристалле для сопряжения вычислительных ядер с кэшем третьего уровня. Принимая во внимание требования к простоте и эффективности реализации таких протоколов, предлагается работать в едином частотном диапазоне для всех радиointерфейсов, используя технологию временного разделения канала с фиксированными временными интервалами. В работе предлагается структура блоков данных, посредством которых идет взаимодействие между узлами. Возможность создать общий счетчик времени обеспечивает синхронизацию между узлами. Временное разделение используется для организации дуплексной связи, причем долю времени, отведенную нисходящему и восходящему каналу, можно изменять динамически. Механизм временного разделения позволяет адаптироваться под неравномерность нагрузки между ядрами посредством выделения различного количества временных слотов на каждое ядро.

### Ключевые слова

процессор общего назначения, иерархическое кэширование, терагерцовый частотный диапазон, беспроводной множественный доступ.

### Благодарности

Работа осуществлена в рамках программы Академии Финляндии «FiDiPro» «Nano Communication Networks», 2012–2016, и частично поддержана Правительством Российской Федерации, грант 074-U01.

## CENTRALIZED MAC PROTOCOL FOR HIERARCHICAL CACHING PROCESSORS

A.A. Antonov<sup>a</sup>, A.O. Kluchev<sup>a</sup>, M.S. Komar<sup>b</sup>, P.V. Kustarev<sup>a</sup>, Ye.A. Koucheryavy<sup>c</sup>,  
D.A. Moltchanov<sup>c</sup>, V.I. Petrov<sup>d</sup>, A.E. Platunov<sup>a</sup>

<sup>a</sup> ITMO University, Saint Petersburg, 197101, Russian Federation

<sup>b</sup> P.G. Demidov Yaroslavl State University, Yaroslavl, 150000, Russian Federation

<sup>c</sup> Tampere University of Technology, Tampere, 33720, Finland

<sup>d</sup> Bonch-Bruевич Saint Petersburg State University of Telecommunications, Saint Petersburg, 193232, Russian Federation

Corresponding author: [platunov@lmt.ifmo.ru](mailto:platunov@lmt.ifmo.ru)

**Article info:**

Received 17.03.15, accepted 27.03.15

doi: 10.17586/2226-1494-2015-15-3-463-469

Article in Russian

**For citation:** Antonov A.A., Kluchev A.O., Komar M.S., Kustarev P.V., Koucheryavy Ye.A., Moltchanov D.A., Petrov V.I., Platunov A.E. Applicability Centralized MAC protocol for hierarchical caching processors. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2015, vol.15, no. 3, pp. 463–469.

**Abstract**

The subject of research is the architecture of modern processors with hierarchical organization of cache subsystem. Analysis of implementation possibility of wireless connectivity between cores and Level 3 cache has been carried out. In order to specify the requirements to wireless communication channel, communication in modern general-purpose processors has been explored by an example of Intel Core i7 (Haswell). Interaction model of cache subsystem components has been developed, and on its basis interaction characteristics between them are being evaluated. Analysis of the model shows that the latency of cache line transmission via the proposed channel is about 0.26 nanoseconds, which correlates well with the latency of L1 cache (about 2 nanoseconds). Also, wireless channel satisfies the distance requirements, giving the possibility for data transmission upto 3 centimeters, as well as power requirements with consumption of 1 uW. The result of research is the developed medium access protocol for wireless connectivity between computational cores and Level 3 cache. To account for required simplicity of implementation and efficiency of operation, it is proposed to use the single frequency range for all radio interfaces and time division multiple access scheme with prescribed fields for addressing and data. The paper deals with protocol data unit structure, which is used for communication between units. Possibility of shared time counter creation is used for synchronization between units. Time division duplex with possibly dynamic non-equal time shares is used to organize uplink and downlink communications. Time division mechanism gives the possibility for the system to adapt to load irregularities between the cores through allocation of various amounts of time slots for each core.

**Keywords**

general-purpose processors, hierarchical cache, terahertz frequency range, wireless multiple access.

**Acknowledgements**

This work was carried out within Academy of Finland FiDiPro program "Nano Communication Networks", 2012–2016, and partially financially supported by the Government of the Russian Federation, Grant 074-U01.

**Введение**

Необходимость увеличения вычислительной мощности компьютеров наблюдается повсеместно. Однако значительно увеличивать для этого частоту работы вычислительных элементов не представляется возможным. Альтернативный подход – создание многоядерных процессоров – зарекомендовал себя при создании требовательных к производительности приложений [1], однако и он не лишен недостатков. Основной из них заключается в сложности масштабирования системы в целом на случай большого количества ядер. Исследования, посвященные этой проблеме, предлагают в качестве решения использование сетей (систем) на кристалле [2]. Они позволяют применить наработки в компьютерных сетях к многопроцессорным системам. Развитие этой идеи привело к появлению концепции беспроводных сетей на кристалле [3], согласно которой некоторое количество расположенных на одном кристалле ядер обмениваются сообщениями друг с другом, используя широкополосные каналы связи.

Однако некоторые проблемы, связанные с применимостью беспроводных сетей на кристалле для реальных процессоров, все еще существуют. Основной из них является организация межъядерного взаимодействия в современных процессорах, которая сейчас представляет собой общение через общую кэш-память верхнего уровня («AMD desktop processor solutions», <http://products.amd.com>; «5th Generation Intel Core Processors Based on the Mobile UProcessor Line», <http://www.intel.com>). Кроме непосредственно обмена данными между ядрами и компонентами подсистемы памяти, наличие коммуникационной инфраструктуры позволяет реализовывать протоколы поддержки когерентности кэш-памяти. Данные протоколы предназначены для поддержания у каждого ядра, использующего какие-либо данные совместно с другими ядрами, актуальной копии данных. Кроме того, требования к задержкам в подобной системе весьма строги – например, согласно документации, время доступа к кэш-памяти третьего уровня в процессорах Intel Sandy Bridge не должно превышать 15 нс [4]. Топологии беспроводных систем на кристалле не всегда могут обеспечить подобную задержку. Кроме того, протоколы когерентности кэш-памяти сложны сами по себе [5], и их наложение на не менее сложные протоколы обмена данными внутри беспроводной системы на кристалле может привести к проблемам с масштабируемостью даже для небольшого числа ядер.

Несмотря на вышесказанное, ключевая идея концепции беспроводных систем на кристалле видится перспективной. Используя эту идею в реальных процессорных архитектурах, можно получить работающие решения, которые позволят обеспечить определенную масштабируемость, при этом обладая достаточно низкой сложностью.

Дальнейшее изложение организовано следующим образом. Рассматривается современная архитектура процессора и архитектура, которую можно получить с применением беспроводных сетей на кристалле. Проведен обзор подсистемы кэш-памяти процессоров Intel Core i7 (Haswell). Предлагается модель взаимодействия компонентов кэш-подсистемы процессора, и выполняется анализ данной подсистемы. По результатам выполненного анализа разработан и представлен протокол множественного доступа.

### Архитектура процессора с иерархическим кэшированием

Современные процессоры общего назначения состоят из большого количества составляющих. Основные компоненты, которые можно выделить в процессоре – это вычислительное ядро с регистрами, кэш-память нескольких уровней и шины данных, соединяющие эти элементы. Подробнее эта схема представлена на рис. 1.

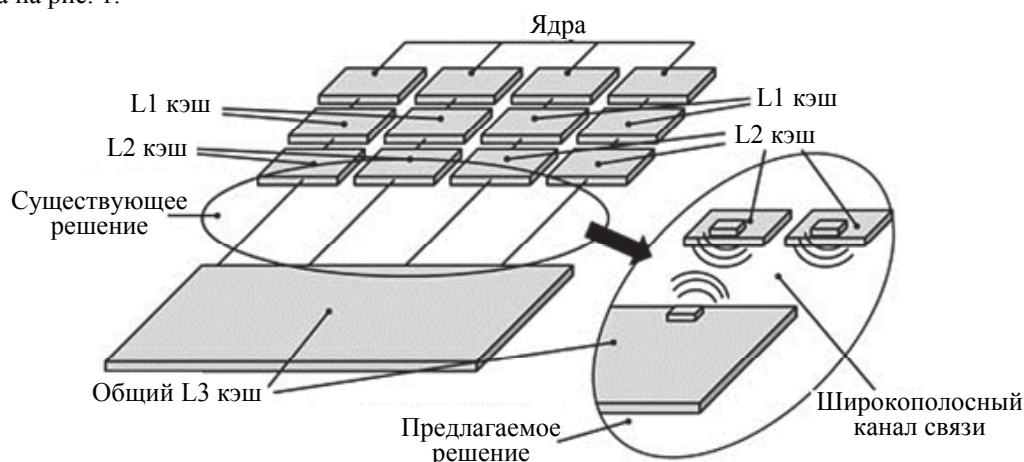


Рис. 1. Существующая и предлагаемая архитектура процессоров (L1, L2, L3 – кэш 1-го, 2-го и 3-го уровня соответственно)

Кэш-память позволяет хранить данные, которые с большой вероятностью понадобятся вычислительным ядрам в ближайшее время. Кроме того, во многих процессорах кэш-память верхнего уровня является общей для нескольких (или всех) ядер [2, 3], что, как было сказано выше, позволяет ядрам обмениваться друг с другом информацией и поддерживать актуальную копию данных, которые используются несколькими ядрами одновременно. Работу с общей кэш-памятью в таком случае организовать достаточно сложно. Например, в случае восьмиядерного процессора ядра разделены на две группы по 4, каждая из которых имеет свою кэш-память третьего уровня. Два блока кэш-памяти верхнего уровня, в свою очередь, соединены между собой высокоскоростной шиной [3]. Такой подход вводит дополнительную задержку на обмен данными между кэш-памятью третьего уровня и также приводит к проблемам с масштабированием системы на большее количество ядер. В таких условиях замена проводной связи между кэш-памятью второго уровня каждого из ядер и общей кэш-памятью третьего уровня, изображенная на рис. 1, представляется перспективным решением. С одной стороны, это не требует серьезных изменений в архитектуре существующего процессора, а значит, уменьшает риски и стоимость модификации. С другой стороны, это позволяет улучшить масштабируемость системы и интегрировать большее количество ядер на одном кристалле. Однако для того, чтобы подобная замена была возможной, необходимо выполнить ряд требований. Во-первых, задержки на доступ к кэш-памяти третьего (верхнего) уровня и на доступ к основной памяти с произвольным доступом (RAM) не должны превышать существующие значения. В данном случае необходимо учитывать не только задержку на передачу сообщения определенной длины, но и задержки, связанные с протоколом доступа к среде. Во-вторых, пропускная способность канала связи должна быть не меньше, чем у существующей шины.

#### Подсистема кэш-памяти в процессорах Intel

В документации на процессоры Intel («Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volumes»; «Intel® 64 and IA-32 Architectures Optimization Reference Manual») приведены некоторые детали архитектуры подсистемы памяти процессора Intel Core i7 (Haswell). На его примере мы постараемся выделить наиболее значимые особенности подсистемы памяти в разрезе исследования трафика между входящими в нее блоками.

Подсистема памяти Intel Haswell имеет многоуровневую иерархию кэш-памяти, состоящую из блоков частично-ассоциативных кэшей с политикой обновления writeback. «Ближайший» уровень кэш-памяти (кэш L1) также характеризуется политикой write allocate. Это означает, что запись обновленных значений производится в L1, а обновление «удаленных» уровней откладывается. В случае, если искомая строка отсутствует в L1, производится ее запрос, поиск в подсистеме памяти ее наиболее актуального экземпляра, загрузка в L1 и запись нового значения. Размер строки составляет 64 Б.

В архитектуре в рамках каждого ядра предусмотрен отдельный кэш 1-го уровня для инструкций и данных (32 КБ каждый), кэш 2-го уровня (256 КБ), общий для инструкций и данных, и кэш третьего уровня (LLC), разделяемый между ядрами и имеющий многобанковую (multiple slice) организацию (размер варьируется, в нашем случае – 8 МБ). Для ускорения доступа к основной памяти также предусмотре-

на возможность подключения кэша L4. Кэш L3 является инклюзивным, в то время как кэш L2 – нет. Таким образом, если искомой строки нет в L3, ее не может быть в L1 или L2 какого-либо ядра. Напротив, отсутствие строки в L2 не гарантирует отсутствие строки в L1.

Ориентировочные характеристики блоков в составе иерархии кэш-памяти могут быть найдены в документации на устройство. Некоторые значения для Intel Haswell приведены в таблицы.

Уровень	Минимальная задержка, такты	Пиковая пропускная способность, (Б/такт)
L1 (данные)	4	64 (чтение) + 32 (запись)
L2 (общий)	11	64

Таблица. Характеристики блоков кэш-подсистемы в порядке поиска (lookup order)

Порядок поиска дополняется поиском в L1 и L2 других ядер, а также, в последнюю очередь, в основной памяти RAM.

Стоит отметить, что приведенные характеристики являются неполными и представлены как наилучшие, что делает актуальной задачу измерения реально достижимых характеристик в рамках натурных испытаний. Выявление более точных характеристик производительности компонентов кэш-подсистемы является предметом будущего исследования.

### Модель взаимодействия между кэш-памятью второго и третьего уровней

В данном разделе, исходя из доступной информации, сформирована модель поведения блоков иерархии памяти при их взаимодействии с процессорным ядром.

При обращении процессорного ядра к подсистеме памяти возможны следующие случаи (в порядке поиска строки):

1. строка данных находится в L1 процессорного ядра-инициатора запросов;
2. строка данных находится в L2 процессорного ядра-инициатора запросов;
3. строка данных находится в L3;
4. строка данных находится в L1/L2 соседнего ядра;
5. строка данных находится в RAM.

Поскольку L2 не является инклюзивным, то поиск в кэш-подсистеме соседнего ядра требует обращений как к L1, так и L2, поэтому данные случаи были объединены. Кроме того, исходя из политики функционирования кэш-подсистемы, ситуации чтения и записи при промахе по L1 предусматривают загрузку строки в L1 с последующим обращением к L1 по чтению или записи, так что ситуации чтения и записи также объединяются. Таким образом, с позиции исследования трафика на шине L2-L3 интерес представляют случаи 2–5. При построении временных диаграмм для каждого из случаев мы исходили из нижеследующих предположений.

Поскольку структура кэш-подсистемы предусматривает соединение каждого последующего уровня с предыдущим, а не с централизованным коммутатором, логично предположить, что поиск в кэшах различных уровней идет последовательно согласно порядку поиска.

При выборке данных из кэша заданного уровня в L1 мы подразумеваем, что либо в L1 есть свободное место для новой строки, либо вытесняемая строка не изменялась. В ином случае чтение сопровождается сохранением вытесняемой строки в кэше L2 и, далее, если и там свободного места нет, в L3.

Под вопросом стоит соотношение времени, затрачиваемого на поиск данных в частично-ассоциативном кэше, а также на поиск плюс выборку при обнаружении данных. Параллельное считывание всех банков памяти тегов и памяти данных делает это время идентичным, но ценой высокого энергопотребления. Однако, поскольку для кэш-памяти высоких уровней (как, например, L1) на первом месте стоит снижение задержки и характерной является параллельная выборка [6], а в других случаях возможно применение сложных схем предсказания [7, 8], позволяющих практически достичь показателей параллельного доступа [9], будем считать эти времена идентичными.

Поскольку запрос и ответ в высокопроизводительных сетях на кристалле расщепляются, а пакеты дублируются и переупорядочиваются, пакеты данных могут аннотироваться идентификаторами транзакции и порядкового номера пакета. Этот подход можно часто встретить как в академических [10], так и промышленных протоколах (например, AMBA AXI, см. «AMBA AXI and ACE Protocol Specification», [www.arm.com](http://www.arm.com)). Однако поскольку длина идентификаторов имеет логарифмическую зависимость относительно количества возможных транзакций, идущих параллельно, для размеров этих полей характерен порядок единиц битов, так что этими полями при расчете трафика можно также пренебречь.

Временные диаграммы для четырех представленных случаев показаны на рис. 2. Трафик на шине L2-L3 выделен жирными линиями.

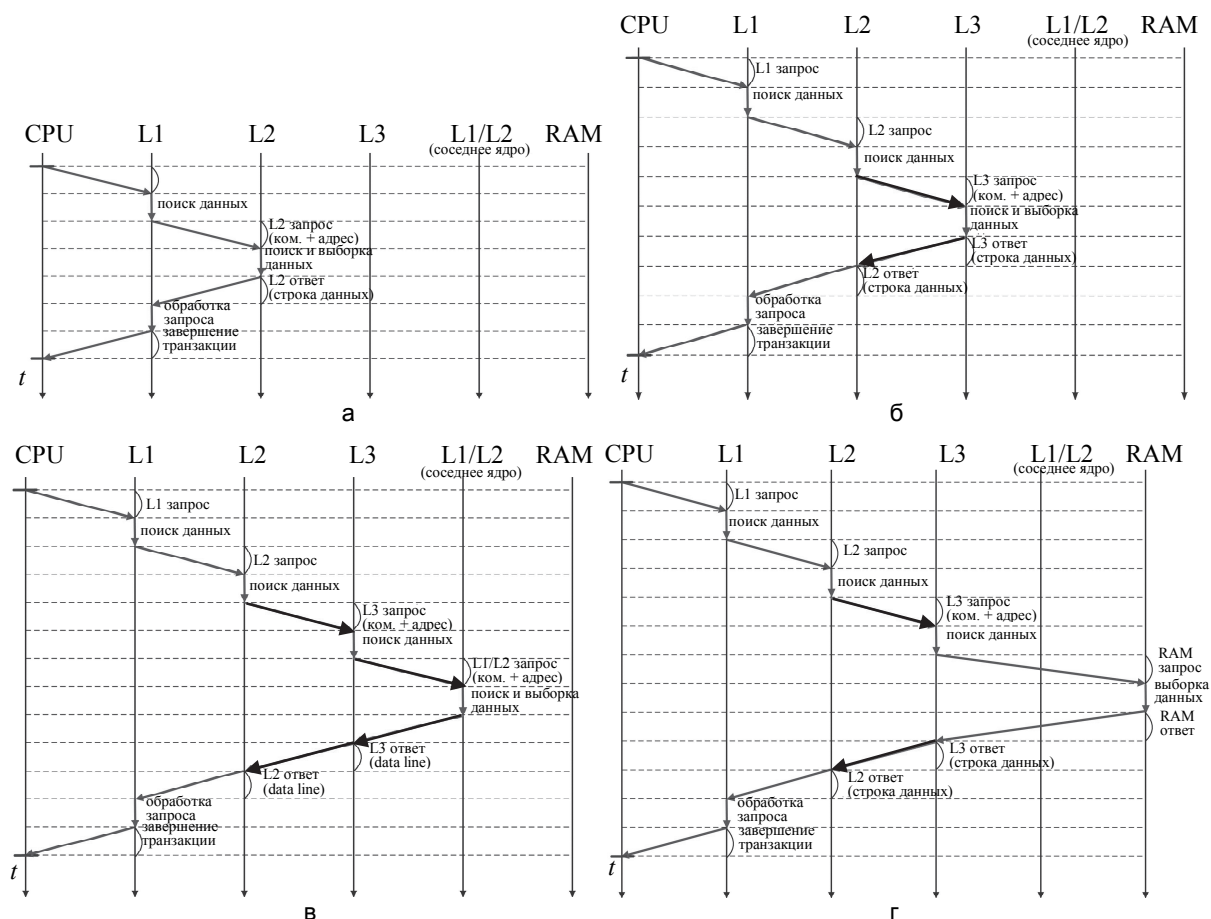


Рис. 2. Временные диаграммы взаимодействия процессорного ядра с компонентами подсистемы памяти: данные в L2 (а); данные в L3 (б); данные в L1/L2 соседнего ядра (в); данные в RAM (г)

### Протокол множественного доступа к среде передачи

В данной работе рассматривается широкополосный беспроводной интерфейс связи между ядрами в частотном диапазоне 0,1–10 ТГц. Согласно имеющимся оценкам, емкость такого канала связи может достигать десятков терабит в секунду на дистанциях в несколько миллиметров [11]. Используя поддиапазон частот 0,1–0,54 ТГц и всего лишь  $1 \cdot 10^{-19}$  Дж энергии на символ, можно достигнуть пропускной способности в 2 Тбит/с на дистанциях до 3 см [12], что сопоставимо с физическим размером современных процессоров. Если учитывать только затраты на передачу непосредственно данных, то одна линия кэша в 64 Б может быть передана по терагерцовому каналу всего за 0,26 нс, что даже меньше, чем задержка при обращении к кэшу первого уровня в существующих процессорах (около 2 нс). Вдобавок к высокой емкости, терагерцовый канал связи также отличается потенциально высоким уровнем энергоэффективности передатчиков и приемников сигнала, расходуя порядка 1 мкВт [11]. С учетом указанных особенностей терагерцового канала связи далее в этом разделе мы предлагаем протокол беспроводного множественного доступа к среде для ядер процессора.

Поскольку речь идет о детерминистическом размещении фиксированного числа узлов и существует возможность создать общий счетчик времени, с высокой степенью достоверности можно предполагать, что время синхронизировано между всеми узлами сети. Для упрощения дальнейшего анализа также предположим, что все ядра процессора имеют примерно равную загрузку. Далее допускаем, что все беспроводные интерфейсы работают в одном и том же частотном диапазоне. Это означает, что невозможна одновременная успешная передача данных между двумя парами узлов. На основе введенных предположений далее разработан протокол множественного доступа к среде на основе временного разделения канала.

В работе [13] было показано, что протокол множественного доступа на основе фиксированного расписания применительно к решаемой задаче обладает существенными преимуществами по сравнению с протоколом случайного доступа. Также было установлено, что емкость исследуемого канала связи достаточна для корректной работы системы с фиксированным расписанием. Следовательно, система не работает в режиме перегрузки [14] и стабильна во времени.

Временная диаграмма протокола изображена на рис. 3. Все время работы системы предлагается разделить на промежутки равной длительности (МакроСлоты). Каждый МакроСлот, в свою очередь, состоит из слота в нисходящем канале связи (от кэша третьего уровня к ядрам) и восходящем канале связи

(от ядер к кэшу третьего уровня). В нисходящем канале связи кэш третьего уровня в эксклюзивном режиме передает массив из  $Q$  блоков данных, каждый из которых содержит в себе 8 Б адреса и 64 Б непосредственно данных. Ресурс восходящего канала, в свою очередь, равномерно разделен между  $N$  ядрами процессора, каждое из которых получает свой слот времени на передачу. В этот слот времени ядро сначала имеет возможность передать запрос на чтение данных из памяти кэша третьего уровня, из кэшей других ядер или оперативной памяти компьютера (ОП). Для этого используется блок из 10 Б, состоящий из двухбайтового идентификатора ядра (в простейшем случае – номера ядра) и восьмибайтного адреса запрашиваемых данных. Затем, не прекращая передачу, ядро имеет возможность записать в кэш третьего уровня или в ОП один блок данных с результатами своих вычислений. Используется блок данных длиной 74 Б: 2 Б на идентификатор ядра, 8 Б на адрес, по которому нужно произвести запись, и 64 Б непосредственно на данные. После успешной передачи данных беспроводной интерфейс ядра отключается до начала следующего МакроСлота.

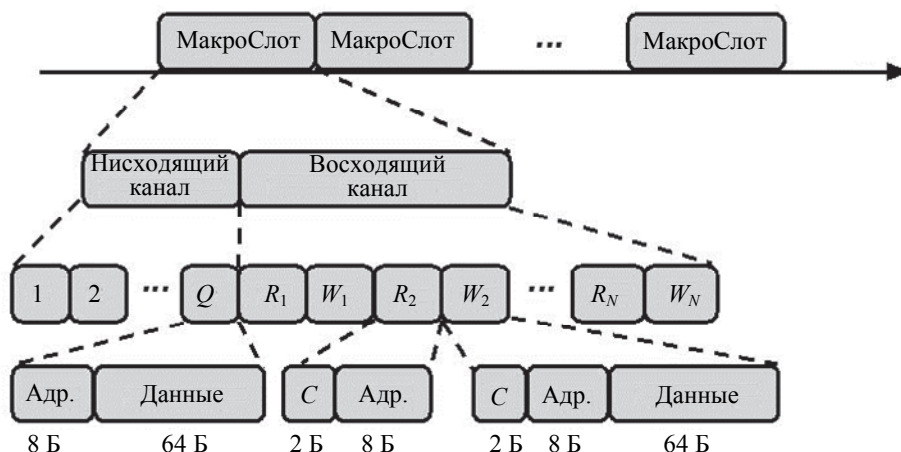


Рис. 3. Протокол доступа к среде ( $Q$  – количество блоков в нисходящем канале связи;  $R_i$  – запрос на чтение;  $W_i$  – запрос на запись;  $N$  – количество ядер;  $C$  – номер ядра)

В общем случае, при неравномерной нагрузке между ядрами, вместо одного слота на передачу данных на каждый МакроСлот ядру может выдаваться несколько или ни одного слота, в зависимости от количества создаваемых обращений на чтение/запись. Соотношение между восходящим и нисходящим каналами настраивается мотивированным выбором параметра  $Q$  – числа блоков данных в нисходящем канале на один МакроСлот.

### Заключение

В рамках работы предлагается модель взаимодействия кэш-памяти второго и третьего уровней в составе многопроцессорной системы с иерархической организацией кэш-памяти. На основе анализа модели в работе представлен протокол множественного доступа к беспроводной среде для ядер такой системы. Помимо множественного доступа, предложенный протокол использует временное разделение для организации дуплексного обмена. Долю времени, отводимую нисходящему и восходящему каналам, предлагается выделять динамически. Метод расчета оптимального значения параметров предлагаемого протокола, а также оценка зависимости между значениями этих параметров и метриками качества обслуживания (например, характеристик задержки в системе) представляет собой задачу дальнейшего исследования.

### Литература

1. Martin C. Multicore processors: challenges, opportunities, emerging trends // Proc. Embedded World Conference 2014. Nuremberg, Germany, 2014. P. 1–9.
2. Bjerregaard T., Mahadevan S. A survey of research and practices of network-on-chip // ACM Computing Surveys. 2006. V. 38. N 1. P. 71–121.
3. Chang K., Deb S., Ganguly A., Yu X., Sah S.P., Pande P.P., Belzer B., Heo D. Performance evaluation and design trade-offs for wireless network-on-chip architectures // ACM Journal of Emerging Technologies in Computing Systems. 2012. V. 8. N 3. Art. 23. doi: 10.1145/2287696.2287706
4. Molka D., Hackenberg D., Schone R. Main memory and cache performance of Intel Sandy Bridge and AMD Bulldozer // Proc. Workshop on Memory Systems Performance and Correctness (MSPC 2014). Edinburgh, UK, 2014. Art. 4. doi: 10.1145/2618128.2618129
5. Sorin D.J., Hill M.D., Wood D.A. A primer on memory consistency and cache coherence // Synthesis Lectures on Computer Architecture. 2011. V. 16. P. 1–212. doi: 10.2200/S00346ED1V01Y201104CAC016

6. Bardizbanyan A., Sjölander M., Whalley D., Larsson-Edefors P. Reducing set-associative L1 data cache energy by early load data dependence detection (ELD<sup>3</sup>) // Proc. 17<sup>th</sup> Design, Automation and Test in Europe (DATE 2014). Dresden, Germany, 2014. Art. 6800296. doi: 10.7873/DATE2014.095
7. Sleiman F.M., Dreslinski R.G., Wenisch T.F. Embedded way prediction for last-level caches // Proc. IEEE 30<sup>th</sup> Int. Conf. on Computer Design. Montreal, Canada, 2012. Art. 6378636. P. 167–174. doi: 10.1109/ICCD.2012.6378636
8. Chu Y., Park J.H. Dual-access way-prediction cache for embedded systems // EURASIP Journal on Embedded Systems. 2014. V. 2014. Art. 16. doi: 10.1186/1687-3963-2014-16
9. Powell M., Agarwal A., Vijaykumar T.N., Falsafi B., Roy K. Reducing set-associative cache energy via way-prediction and selective direct-mapping // Proc. 34<sup>th</sup> Annual International Symposium on Microarchitecture ACM/IEEE. Austin, USA, 2001. P. 54–65. doi: 10.1109/MICRO.2001.991105
10. Tota S.V., Casu M.R., Roch M.R., Rostagno L., Zamboni M. Medea: a hybrid shared-memory/message-passing multiprocessor NoC-based architecture // Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE 2010). Dresden, Germany, 2010. Art. 5457237. P. 45–50.
11. Jornet J.M., Akyildiz I.F. Channel modeling and capacity analysis for electromagnetic wireless nanonetworks in the terahertz band // IEEE Transactions on Wireless Communications. 2011. V. 10. N 10. P. 3211–3221. doi: 10.1109/TWC.2011.081011.100545
12. Boronin P., Petrov V., Moltchanov D., Koucheryavy Y., Jornet J.M. Capacity and throughput analysis of nanoscale machine communication through transparency windows in the terahertz band // Nano Communication Network. 2014. V. 5. N 3. P. 72–82. doi: 10.1016/j.nancom.2014.06.001
13. Комар М.С., Кучерявый Е.А., Молчанов Д.А., Петров В.И. Расчет характеристик протоколов беспроводной связи для взаимодействия между ядрами центрального процессора // Информационные технологии и телекоммуникации. 2014. № 3. С. 41–58.
14. Petrov V., Andreev S., Turlikov A., Koucheryavy Y. On IEEE 802.16m overload control for smart grid deployments // Lecture Notes in Computer Science. 2012. V. 7469 LNCS. P. 86–94. doi: 10.1007/978-3-642-32686-8\_8

<i>Антонов Александр Александрович</i>	– студент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, antonov.alex.alex@gmail.com
<i>Ключев Аркадий Олегович</i>	– кандидат технических наук, доцент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, kluchev@lmt.ifmo.ru
<i>Комар Мария Сергеевна</i>	– студент, Ярославский государственный университет им. П.Г. Демидова, Ярославль, 150000, Российская Федерация, maria.s.komar@gmail.com
<i>Кустарев Павел Валерьевич</i>	– кандидат технических наук, доцент, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, kustarev@lmt.ifmo.ru
<i>Кучерявый Евгений Андреевич</i>	– кандидат наук, профессор, профессор, Технологический университет Тампере, Тампере, 33720, Финляндия, yk@cs.tut.fi
<i>Молчанов Дмитрий Александрович</i>	– кандидат наук, доцент, доцент, Технологический университет Тампере, Тампере, 33720, Финляндия, dmitri.moltchanov@tut.fi
<i>Петров Виталий Игоревич</i>	– аспирант, Санкт-Петербургский университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация, vit.petrov@gmail.com
<i>Платунов Алексей Евгеньевич</i>	– доктор технических наук, профессор, профессор, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, platonov@lmt.ifmo.ru
<i>Alexander A. Antonov</i>	– student, ITMO University, Saint Petersburg, 197101, Russian Federation, antonov.alex.alex@gmail.com
<i>Arkady O. Kluchev</i>	– PhD, Associate professor, ITMO University, Saint Petersburg, 197101, Russian Federation, kluchev@lmt.ifmo.ru
<i>Maria S. Komar</i>	– student, P.G. Demidov Yaroslavl State University, Yaroslavl, 150000, Russian Federation, maria.s.komar@gmail.com
<i>Pavel V. Kustarev</i>	– PhD, Associate professor, ITMO University, Saint Petersburg, 197101, Russian Federation, kustarev@lmt.ifmo.ru
<i>Yevgeni A. Koucheryavy</i>	– PhD, Professor, Professor, Tampere University of Technology, Tampere, 33720, Finland, yk@cs.tut.fi
<i>Dmitri A. Moltchanov</i>	– PhD, Associate professor, Associate professor, Tampere University of Technology, Tampere, 33720, Finland, dmitri.moltchanov@tut.fi
<i>Vitaly I. Petrov</i>	– postgraduate, Bonch-Bruевич Saint Petersburg State University of Telecommunications, Saint Petersburg, 193232, Russian Federation, vit.petrov@gmail.com
<i>Alexey E. Platonov</i>	– D.Sc., Professor, Professor, ITMO University, Saint Petersburg, 197101, Russian Federation, platonov@lmt.ifmo.ru