

А.С. Кожин (ЗАО «МЦСТ»), **Ю.А. Недбайло** (ЗАО «МЦСТ», ПАО «ИНЭУМ им. И.С. Брука»)

A. Kozhin, Y. Nedbailo

ОПТИМИЗАЦИЯ ОБЩЕГО КЭША ТРЕТЬЕГО УРОВНЯ МИКРОПРОЦЕССОРА «ЭЛЬБРУС-8С»

OPTIMIZING THE INCLUSIVE SHARED L3 CACHE IN «ELBRUS-8C» MICROPROCESSOR

Описана оптимизация инклюзивного общего кэша третьего уровня микропроцессора «Эльбрус-8С» путём реализации инклюзивного справочника и перевода самого кэша на принцип неинклюзивности. Приведена оценка прироста производительности для разных режимов неинклюзивности на основе моделирования. Для предложенного решения оценены затраты площади и энергопотребления.

The paper describes inclusive shared L3 cache optimization of the Elbrus-8C microprocessor through implementation of inclusive directory and making the cache itself non-inclusive. Performance gain for different non-inclusiveness modes is evaluated using trace-driven simulation. Power and area costs of the proposed solution are estimated.

Ключевые слова: микропроцессор «Эльбрус», архитектура, многоядерность, подсистема памяти, общий кэш третьего уровня, инклюзивность, моделирование.

Keywords: Elbrus microprocessor, architecture, many-core, memory subsystem, shared L3 cache, NCID, simulation.

Введение

Развитие полупроводниковой микроэлектроники позволяет примерно раз в два года удваивать количество оборудования, приходящегося на единицу площади,

энергопотребления и цены. В многоядерных процессорах это способствует наращиванию производительности путём увеличения количества ядер. Но, для того чтобы этот рост реализовался в широком спектре задач, требуется пропорциональное увеличение пропускной способности подсистемы памяти. Однако оперативная память эволюционирует не такими быстрыми темпами: смена поколений интерфейсов DDR SDRAM, приносящая удвоение пропускной способности, происходит приблизительно раз в четыре года.

Вследствие такого отставания, чтобы поддержать высокий темп роста производительности, требуется всё более экономно использовать пропускную способность памяти, уменьшая среднее количество обращений в память за операцию. Важнейшую роль при этом играет снижение частоты промахов в кэши за счет усовершенствования организации кэш-памяти микропроцессора. В данной статье эта проблема решается на примере восьмиядерного микропроцессора «Эльбрус-8С» путем оптимизации инклюзивного общего кэша третьего уровня (L3-кэша).

1. Известные методы

Эффективность общего кэша, одновременно обслуживающего несколько процессорных ядер, может снижаться из-за нехватки ассоциативности (thrashing), бесполезного вытеснения одним потоком данных, нужных другим потокам (pollution), неравномерного распределения запросов по банкам и сетам вследствие использования недостаточно хорошей функции индексирования. Для борьбы с этими эффектами применяются соответствующие методы. В этой статье рассмотрим оптимизацию инклюзивного кэша, повышающую его эффективность даже при симметричной нагрузке; чтобы показать это, моделирование будет проводиться на 1/8 части кэша, предоставленной одному ядру.

Инклюзивный общий кэш используется в качестве справочника для фильтрации когерентных запросов. Каждая строка, содержащаяся в вышестоящих (приватных) кэшах, каждый из которых связан с одним ядром, обязательно присутствует в общем кэше, поэтому когерентный запрос не нужно отправлять в вышестоящие кэши, если в справочнике общего кэша они не указаны для запрашиваемой строки. Это избавляет от резкого роста трафика когерентных запросов при увеличении количества ядер, но ограничивает объемом общего кэша диапазон адресов, охватывающий все кэши. Соответственно, оптимизации инклюзивного кэша можно разделить на две категории:

- оптимизация замещения в кэше;
- введение инклюзивного справочника при неинклюзивной схеме самого кэша.

Оптимизации первого типа детально рассмотрены в [1], на рис. 1 показана относительная производительность неинклюзивной и эксклюзивной схем по сравнению с инклюзивной в зависимости от соотношения суммарного объема приватных кэшей и объема общего кэша. Как можно заметить, объем инклюзивного кэша достаточно сильно влияет на производительность: при соотношении 1:4 он проигрывает в среднем 8% двум другим схемам, при соотношении 1:8 отставание снижается до 3%. На рисунке приведено также влияние нескольких оптимизаций замещения. Две из них – TLH и QBS – заключаются в предоставлении нижестоящему кэшу информации о состоянии строк в вышестоящих и практически полностью компенсируют негативное влияние инклюзивности на производительность, но относительно сложны в реализации. Третья оптимизация – ECI – достаточно проста для внедрения, но и эффект более скромный – при соотношении 1:4 он составляет около 3,5%.

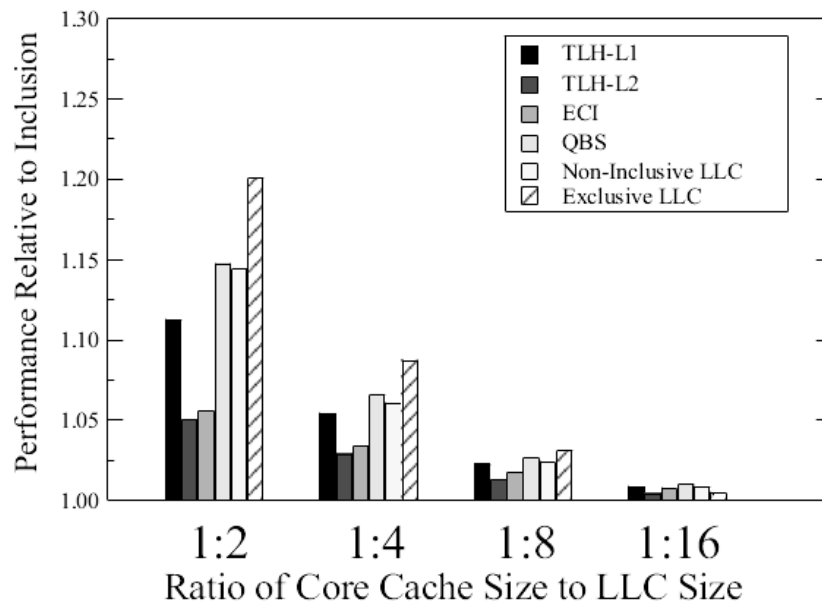


Рис. 1. Прирост производительности относительно инклюзивной схемы при различных оптимизациях в зависимости от соотношения объёмов кэшей [1]

Метод NCID – Non-inclusive Cache, Inclusive Directory – описан в [2], его основная идея состоит в отделении справочника от памяти данных инклюзивного кэша. Сохраняя инклюзивность справочника, он обеспечивает и фильтрацию когерентных запросов, а отказ от строгих требований к размещению данных позволяет снизить негативный эффект инклюзивности и повысить производительность. В данной статье описаны три схемы организации NCID (рис. 2):

- с отдельным справочником;
- с отдельными тегами справочника в каждом сете;
- с общими тегами, динамически распределяемыми между справочником и кэшем.

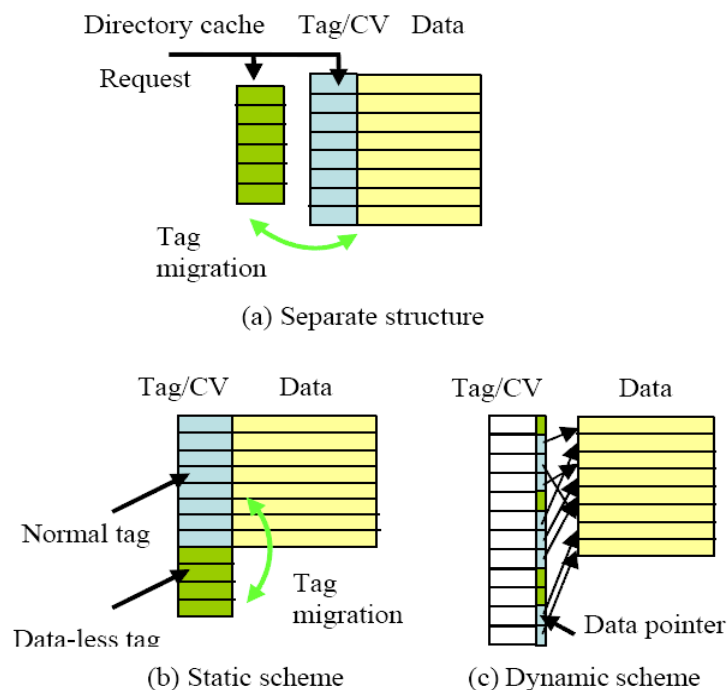


Рис. 2. Схемы структуры неинклюзивного кэша с инклюзивным справочником [2]

По сути, в предложенном методе инклюзивность по тэгам и данным заменяется на инклюзивность только по тэгам. Благодаря этому ее негативное влияние может быть полностью устранено, если справочник представляет собой копию тэгов вышестоящего кэша, хотя такая схема не всегда реализуема из-за микроархитектурных особенностей. В качестве альтернативы копии тэгов авторы статьи предлагают увеличить покрытие справочника (отношение суммарного объема частных кэшей и объема инклюзивного кэша с эквивалентным количеством тэгов) до двух. Следует отметить, что метод NCID значительно сложнее в реализации, чем оптимизации замещения, и может несколько увеличить время доступа в память, но, как будет показано ниже, при этом позволяет добиться большего прироста производительности.

В этой статье мы предлагаем достаточно эффективную структуру NCID первого или второго типа и оценим, насколько неинклюзивность данных может повысить эффективность общего кэша в системе «Эльбрус-8С».

2. Микропроцессор «Эльбрус-8С»

Основные характеристики подсистемы памяти микропроцессора «Эльбрус-8С» [3], структурная схема которого показана на рис. 3, включают:

- 8 ядер с расчетной частотой 1,3 ГГц;
- 4 канала DDR3-1600 SDRAM;
- приватные L1I 128 КБ, L1D 64 КБ и L2 512 КБ 4-way ассоциативные кэши;
- распределённый общий 16 МБ 16-way L3-кэш.

Размер строки L1I кэша 256 Б, у остальных – 64 Б. Протокол когерентности – MOSI.

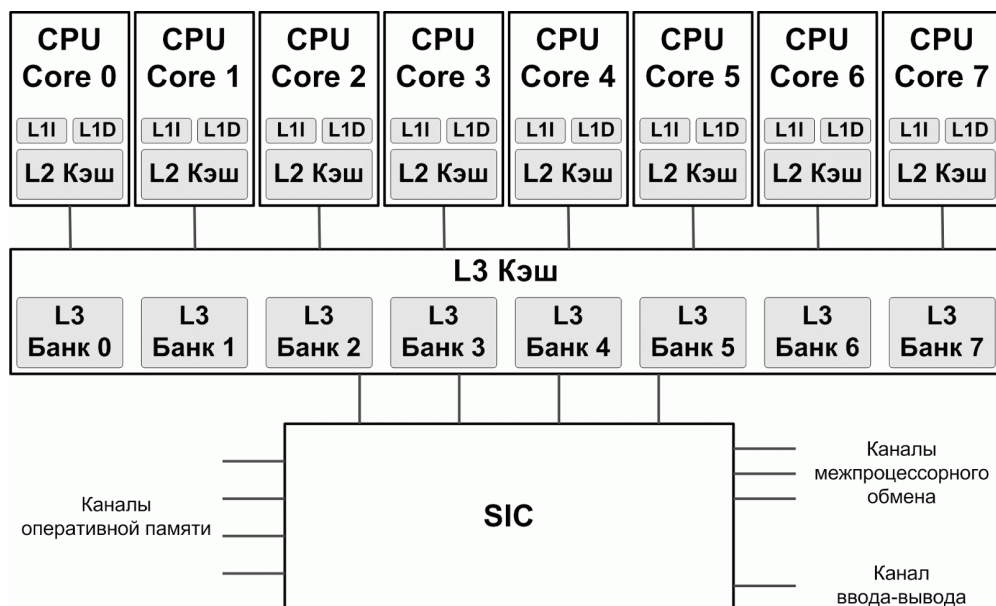


Рис. 3. Структурная схема микропроцессора «Эльбрус-8С»

(CPU Core – процессорное ядро; L1I и L1D – приватные кэши первого уровня для кода и данных соответственно; L2 кэш – приватный кэш второго уровня; L3 кэш – общий кэш третьего уровня; SIC (System Interface Controller) – контроллер системных обменов)

Для моделирования использовались точная программная модель ядра и поведенческие модели кэшей, учитывающие основные детали протокола когерентности и выбранных схем инклюзивности. Модель ядра генерировала трассу обращений программы в подсистему памяти, содержащую номер такта, адрес и тип операции каждого

запроса. Затем трасса подавалась на вход модели кэшей, которая для каждого запроса определяла попадание или промах, соответствующим образом меняла своё состояние и формировала статистику. Таким образом, хотя динамика работы подсистемы памяти моделями не учитывается, и из-за этого частота промахов в кэши получается несколько завышенной, доля промахов и их соотношение между разными схемами предполагаются оцененными достаточно точно, что подтверждается предыдущими модельными экспериментами с микропроцессорами эльбрусовской серии.

3. Неинклюзивный кэш с инклюзивным справочником

Как показано на рис. 1 и подтверждает моделирование, соотношение размеров кэшей $L1D:L2 = 1:8$ достаточно велико, чтобы включение L1D в L2 существенно не повлияло на эффективность. Кэш L1I не обязательно делать когерентным, но если это потребуется, для него можно ввести отдельный небольшой справочник, поскольку размер строки достаточно велик (256 Б) и, соответственно, строк в нём мало. Эти обстоятельства позволяют сделать справочник L3 просто как копии тегов L2, что и наиболее дёшево, и избавит кэши от необходимости гашения строк при вытеснении их из справочника. Исходная инклюзивная и описанная неинклюзивная схемы организации кэшей изображены на рис. 4.

Из возможных типов неинклюзивного кэша нижнего уровня были проанализированы три:

- с заведением по запросам «чтение» и обратной записи из L2 (NI-RA);
- с заведением только для обратной записи из L2 (NI-WB);
- эксклюзивный-victim – заведение при вытеснении из L2 даже немодифицированных, гашение любым другим запросом (EX).

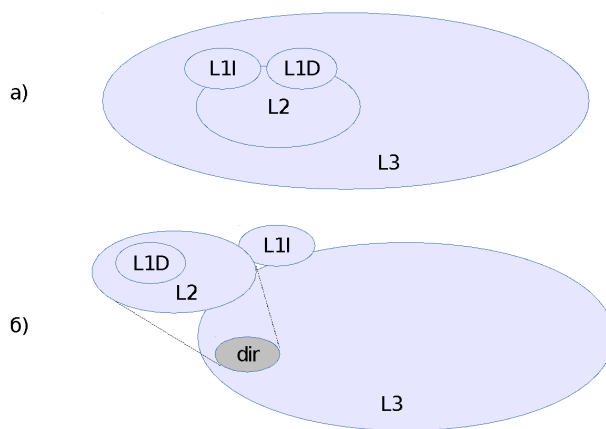


Рис. 4. Охват адресного пространства: а) инклюзивной схемой; б) неинклюзивной

Диаграммы состояний справочника и L3 кэша каждой из этих схем, соответствующих протоколу когерентности MOSI, показаны на рис. 5.

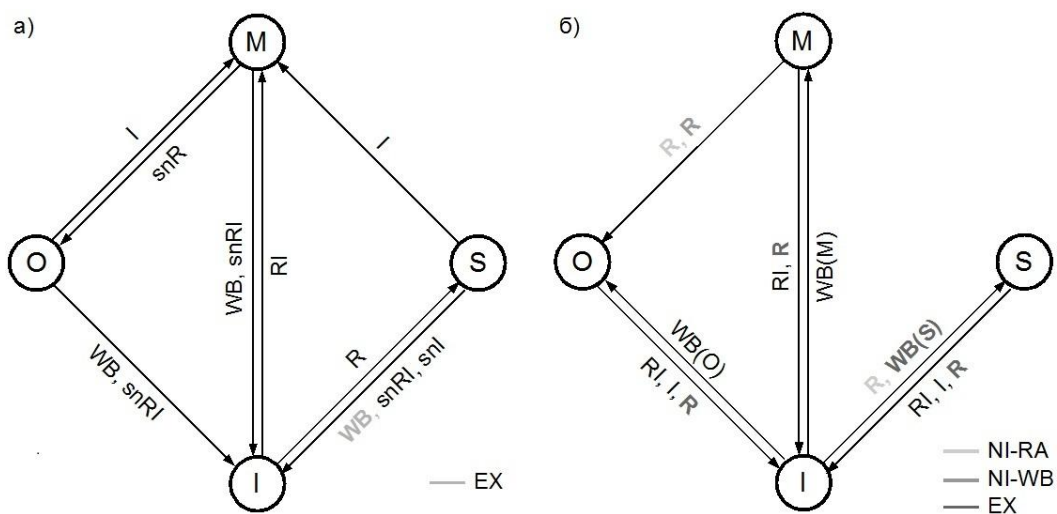


Рис. 5. Диаграммы состояний: а) справочника б) L3 кэша.

В качестве дальнейшей оптимизации предполагалось сделать автоматическое переключение между NI-RA и EX схемами, известное как FLEXclusion и описанное в [4]. Однако результаты моделирования, приведённые на рис. 6, показали, что сложности реализации эксклюзивной схемы в данном случае себя не оправдывают, и переключение лучше и проще реализовать между NI-RA и NI-WB (результаты полученной адаптивной схемы обозначены как NI-FL).

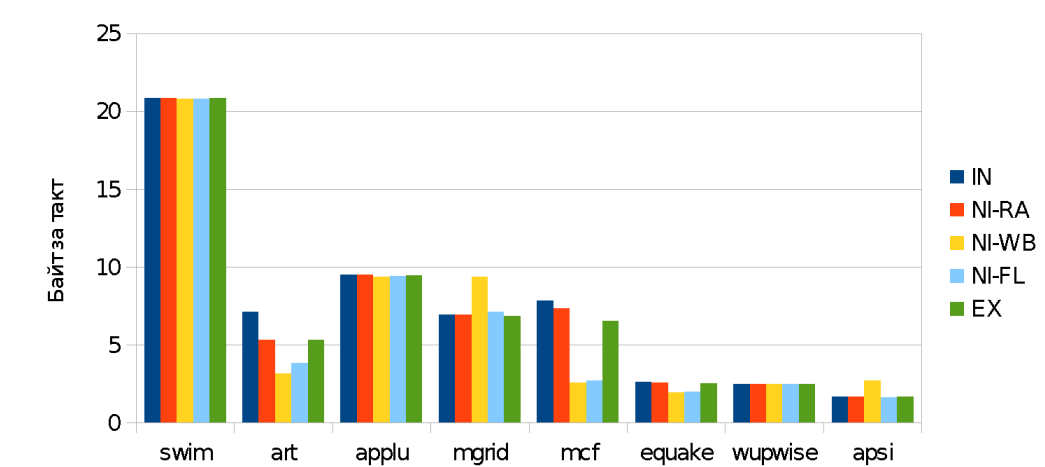


Рис. 6. Частота промахов в L3 кэш

Для переключения между режимами используются два тестовых набора сетов: один, всегда работающий в одном режиме, второй – в другом, и четыре счётчика обращений и промахов в эти сеты. Когда счётчики достигают некоторого порогового значения, подобранного экспериментально, основная часть кэша переключается в режим, для которого доля промахов была меньше, и счётчики сбрасываются.

Как показано на рис. 6, на трассах обращений в память восьми тестов, самых требовательных к пропускной способности подсистемы памяти пакета SPEC CPU2000, обычная неинклюзивная (NI-RA) схема показала в среднем на 5% меньше промахов, чем инклюзивная, а адаптивная неинклюзивная (NI-FL) – на 22% меньше. Полученные результаты коррелируют с данными из исследования [1] и, следовательно, могут учитываться при дальнейшем развитии организации общей кэш-памяти в микропроцессорах серии «Эльбрус». Однако стоит обратить внимание, что проведенное моделирование включает не все особенности микроархитектуры, поэтому возможно количественное расхождение модели с ее аппаратной реализацией.

4. Энергопотребление и площадь

Основной вклад в увеличение площади и рассеиваемой мощности предложенной

схемы вносят дополнительные тэги справочника и схема их проверки. Оценить добавляемую площадь можно, зная, что тэги кэша L3 занимают 19% от его площади, а сам кэш L3 – примерно 30% от общей. Реализация справочника в виде копий тэгов L2 требует 1/4 от количества тэгов L3, и его площадь можно соответственно аппроксимировать как 5% от площади L3 и $1/4 \times 0,19 \times 0,3 \approx 1,5\%$ от общей площади процессора. Если учесть, что часть тэгов L3 занимает ненужная неинклюзивной схеме информация о наличии строки в ядрах, оценка может быть ещё значительно уменьшена. Если делать справочник не привязанным к структуре L2 кэша, он должен быть несколько больше – как показало моделирование, двукратного запаса достаточно, и для этого варианта верхняя оценка будет 10% от площади кэша L3 или 3% от общей площади.

Энергопотребление справочника можно оценить по спецификациям статических памятей и логических элементов, нужных для хранения и сравнения тэгов. Каждый запрос должен считывать $4 \times 8 = 32$ тэга (длиной примерно по 3 байта) из памятей глубиной 256 или 512 для большего варианта и сравнивать их со своим адресом. Для предполагаемого техпроцесса суммарная оценка получается около 1 Вт на все 8 банков. Реальная мощность будет кратно меньше в соответствии с темпом обменов между L2 и L3; его оценка на программной модели показана на рис. 7. Полная мощность достигалась бы при темпе в одну кэш-строку (64 байта) за такт, соответственно, в самых тяжёлых тестах она ожидается в пределах 0,2–0,4 Вт, что составит менее 1% от мощности всего процессора.

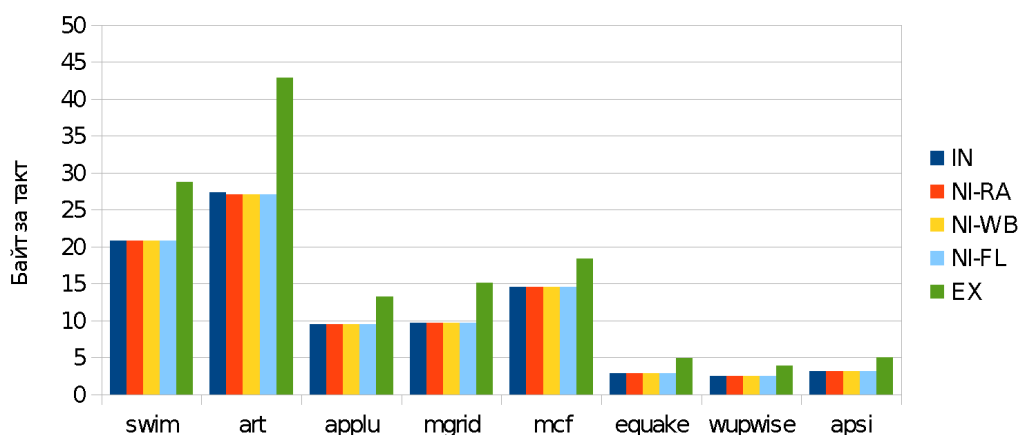


Рис. 7. Частота обменов между L2 и L3 кэшами

Заключение

Описанная в данной статье адаптивная неинклюзивная схема L3 кэша с инклюзивным справочником демонстрирует на симуляторе процессора «Эльбрус-8С» в среднем около 20% снижения частоты промахов по сравнению с инклюзивной схемой. Стоимость этого решения в указанной системе оценена в 1,5–3% от общей площади процессора и 1% мощности. Таким образом, предложенная схема является целесообразной для дальнейшего исследования в части оптимальной организации справочника и более точного тестирования на прототипе [5] с целью внедрения в будущие многоядерные микропроцессоры с архитектурой «Эльбрус».

Литература

1. Aamer Jaleel, Eric Borch, Malini Bhandaru, Simon C. Steely Jr., Joel Emer. Achieving non-inclusive cache performance with inclusive caches: Temporal locality aware (TLA) cache management policies // Proceedings of the 43rd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO'43, Washington, DC, USA. – 2010. – С. 151–162.
2. Li Zhao, Ravi Iyer, Srihari Makineni, Don Newell, Liqun Cheng. NCID: A noninclusive cache, inclusive directory architecture for flexible and efficient cache hierarchies // Proceedings of the 7th ACM International Conference on Computing Frontiers, CF'10, New York, NY, USA. – 2010. – С. 121–130.
3. Кожин А.С., Кожин Е.С., Костенко В.О., Лавров А.В. Кэш третьего уровня и поддержка когерентности микропроцессора «Эльбрус-4С+». – «Вопросы радиоэлектроники», сер. ЭВТ, 2013, вып. 3.
4. Jaewoong Sim, Jaekyu Lee, Moinuddin K. Qureshi, Hyesoon Kim. FLEXclusion:

Balancing cache capacity and on-chip bandwidth via flexible exclusion. – SIGARCH Comput. Archit. News. – 2012. – 40(3):321–332.

5. Будылин Ф.К., Полищук И.А., Слесарев М.В., Юрлин С.В. Опыт прототипирования микропроцессоров компании ЗАО «МЦСТ». – «Вопросы радиоэлектроники», сер. ЭВТ, 2012, вып. 3, с. 132–142.