Для цитирования: Поляков Н. Ю. Трансляция виртуальных адресов DMA-обращений в МП «Эльбрус-8С2» // Вопросы радиоэлектроники. 2017. № 3. С. 22–26.
УДК 004.318

Н. Ю. Поляков[1]

[1] АО «МЦСТ»

ТРАНСЛЯЦИЯ ВИРТУАЛЬНЫХ АДРЕСОВ DMA-ОБРАЩЕНИЙ В МП «ЭЛЬБРУС-8С2»

Постоянное увеличение объема оперативной памяти и ужесточение требований к безопасности современных вычислительных средств обусловило поддержку виртуальной адресации DMA-обращений в операционных системах, периферийных интерфейсах и, наконец, в микропроцессорах (МП). В статье рассматривается первая реализация устройства трансляции виртуальных адресов DMA-обращений в МП с архитектурой «Эльбрус» и описываются усовершенствования, введенные для увеличения быстродействия его следующей версии, которая внедрена в МП «Эльбрус-8С2».

Ключевые слова: *IOMMU, DMA, Эльбрус, виртуализация, IOTLB, prefetch.*

Введение

В большинстве современных микропроцессоров реализовано обращение в оперативную память (DMA, Direct Memory Access) по виртуальным адресам [1–5]. Такая возможность предполагает наличие в аппаратуре устройства трансляции виртуальных адресов в физические IOMMU (I/O Memory Management Unit), выполняемой подобно трансляции при обращениях в память от процессорных ядер. Этот механизм имеет ряд преимуществ перед прямыми обращениями по физическим адресам [3]:

- Разрядность адреса обращения в память, доступная для некоторых внешних устройств, не охватывает всего физического адресного пространства, тогда как введение IOMMU позволяет перенаправлять обращения в области физической памяти, недоступные устройству без трансляции.
- Трансляция адреса позволяет защитить оперативную память от воздействия вредоносных внешних устройств путем перенаправления обращений или запрета обращений к определенным страницам.
- Для DMA-обменов, превышающих по объему одну страницу, может использоваться разделенная на сегменты физическая область, представленная как непрерывная виртуальная, что позволяет устройству вместо цепочки обменов с физической адресацией выполнять один обмен, адресованный к непрерывной виртуальной области.

В то же время механизм трансляции адресов имеет определенные недостатки, обусловленные тем, что часть оперативной памяти используется для хранения таблиц трансляции (PT, Page Table) и снижается производительность подсистемы ввода-вывода за счет добавления дополнительной стадии при обработке DMA-обращения [4]. В связи с этим в МП с архитектурой «Эльбрус» был введен механизм IOMMU, впервые реализованный в МП «Эльбрус-4С» и затем усовершенствованный в МП «Эльбрус-8С2». Данная статья знакомит с принятыми на этих этапах решениями.

IOMMU МП «Эльбрус-4С»

Базовую функциональность устройства IOMMU МП «Эльбрус-4С» (рис. 1) можно рассматривать как первый шаг в виртуализации периферии:

- Поддерживается только одноуровневая таблица страниц.
- Поддерживаются только страницы размером 4 Кб.
- Элемент таблицы страниц содержит поля защиты страницы от записи и/или чтения.
- В составе устройства имеется небольшой кэш элементов таблицы страниц для ускорения трансляции IOTLB (I/O Translation Lookaside Buffer).

В состав IOMMU входят блок обработки запроса на трансляцию (exec station), кэш IOTLB для хранения часто используемых элементов таблицы страниц (PTE, Page Table Element) и блок конфигурационных регистров (CR).

Поступающий для трансляции виртуальный адрес (VA, Virtual Address) помещается на регистровую станцию и находится там вплоть до выдачи физического адреса (PA, Physical Address). В первом такте обработки выполняется поиск по IOTLB. Кэш IOTLB

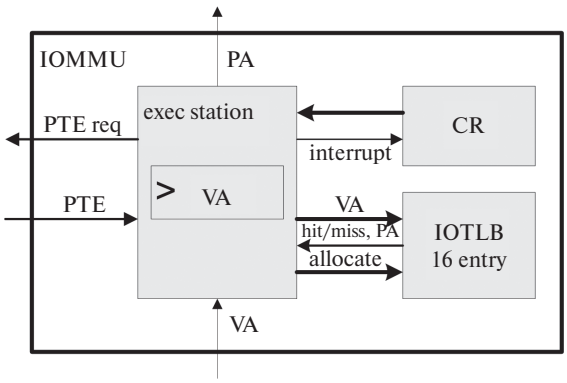

Рисунок 1. Структурная схема IOMMU МП «Эльбрус-4С»

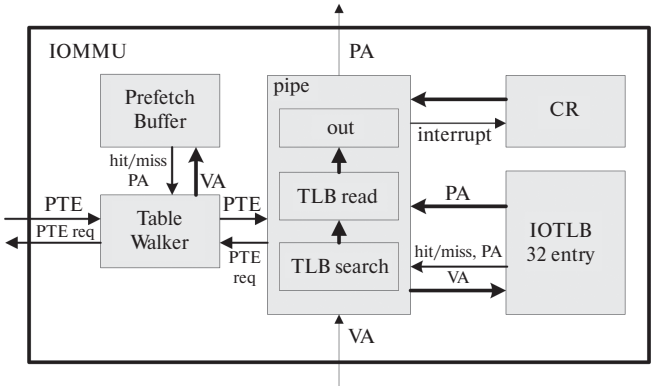

Рисунок 2. Структурная схема IOMMU МП «Эльбрус-8С2»

является полностью ассоциативным и содержит 16 строк. В случае попадания (hit) в этом же такте выполняется чтение физического адреса из буфера, а в следующем – выдается результат трансляции, после чего может быть принят новый запрос. Таким образом, пропускная способность IOMMU в случае попадания составляет один запрос за два такта.

При промахе (miss) формируется запрос на чтение нового элемента таблицы из памяти (PTE req). Он содержит физический адрес и биты разрешения чтения и записи данной страницы внешними устройствами. Если доступ к странице запрещен, выдается прерывание и первичный DMA-запрос завершается. В противном случае физический адрес страницы выдается в ответ на запрос, а PTE помещается в IOTLB. Выбор строки для замещения выполняется по алгоритму FIFO, т. е. замещается тот элемент, который был заведен раньше остальных.

IOMMU МП «Эльбрус-8С2»

В МП «Эльбрус-8С2» устройство IOMMU (рис. 2) получило дальнейшее развитие. Для увеличения производительности были введены следующие усовершенствования:

- Трансляция адреса разбита на три стадии конвейера, в котором каждая стадия выполняется за один такт в случае попадания в IOTLB.
- Размер IOTLB увеличен в два раза, а память физических адресов заменена на блочную.
- В IOMMU введена функциональность по предварительной подкачке PTE из памяти.

В состав устройства входят: конвейер трансляции адреса (pipe), IOTLB, блок конфигурационных регистров (CR), блок чтения PTE из памяти Table Walker (TW), буфер предварительной подкачки элементов таблицы страниц Prefetch Buffer (PB).

Первым фактором увеличения производительности стала конвейеризация обработки запросов на трансляцию (в предыдущей версии IOMMU запросы выполнялись по одному в течение двух тактов каждый). В новой версии трансляция разбита на три стадии:

1. Поиск адреса в IOTLB.
2. Чтение физического адреса из IOTLB.
3. Выдача результата трансляции.

Данное разбиение в первую очередь позволяет увеличить темп приема запросов до одного запроса за такт в случае попадания в IOTLB. Кроме того, добавление дополнительной стадии 2 (чтение физического адреса) позволяет:

- увеличить тактовую частоту работы устройства;
- использовать для хранения физического адреса блочную память, имеющую меньшие площадь и рассеиваемую мощность по сравнению с массивом регистров;
- в результате значительно увеличить количество строк IOTLB при сохранении полной ассоциативности.

Предварительная подкачка PTE

Одним из самых распространенных методов снижения числа промахов по кэш-памяти является предварительная подкачка строк из оперативной памяти [6]. В ОС Linux предусмотрено два механизма выделения памяти для DMA-обмена [7]:

1. Консистентный (или когерентный) – долгосрочное выделение памяти, при котором память выделяется один раз при инициализации драйвера внешнего устройства и освобождается при прекращении работы драйвера. Такой механизм используется, например, для дескрипторов кольцевых буферов сетевой карты.
2. Потоковый – краткосрочное выделение, при котором память предоставляется только для

одного обмена и освобождается сразу после его завершения. Такой механизм используется для буферов приема или передачи данных через сетевую карту и для буферов файловой системы SCSI-устройств. Наиболее эффективно предварительная подкачка PTE работает именно для потоковых DMA-обменов.

В МП «Эльбрус-8С2» реализована предварительная подкачка, основанная на подсказках от драйвера IOMMU, размещаемых в PTE. В зависимости от типа обмена, запрашиваемого драйвером внешнего устройства, драйвер помечает страницу в PTE признаком потоковой (PTE.coh=0) или когерентной (PTE.coh=1) страницы. При получении из памяти PTE с первым признаком IOMMU отправляет в память запрос на подкачку PTE следующей страницы. Подкаченный PTE помещается в буфер подкачки PB. В случае очередного промаха по IOTLB выполняется поиск в PB: при попадании найденный PTE перемещается в IOTLB, иначе запрос отправляется в память. Полный алгоритм работы трансляции с предварительной подкачкой приведен на рис. 3.

Сравнение быстродействия

Так как микроархитектуры МП «Эльбрус-4С» и «Эльбрус-8С2» существенно различны, сравнение быстродействия обменов на этих процессорах некорректно. Вместо этого было проведено сравнение быстродействия обменов на МП «Эльбрус-8С2» и «Эльбрус-8С», имеющем IOMMU первой версии и подсистему памяти, близкую к «Эльбрус-8С2».

Теоретический прирост быстродействия (величина, обратная времени [8]) при каждой доработке можно оценить следующим образом. Конвейеризация дает теоретический прирост темпа обработки запросов в два раза (увеличение быстродействия на 100%) вследствие уменьшения времени исполнения одной стадии трансляции с двух тактов до одного. Предварительная подкачка предотвращает простой конвейера во время чтения PTE из памяти, которое в МП «Эльбрус-8С» и «Эльбрус-8С2» выполняется приблизительно за 70 тактов. Обмен размером 4 Кб (одна страница) по каналу ввода-вывода с эффективной пропускной способностью 5 Гб/с в процессоре с тактовой частотой 1,2 ГГц выполняется за 960 тактов (1,2 ГГц x 4 Кб / 5 Гб/с).

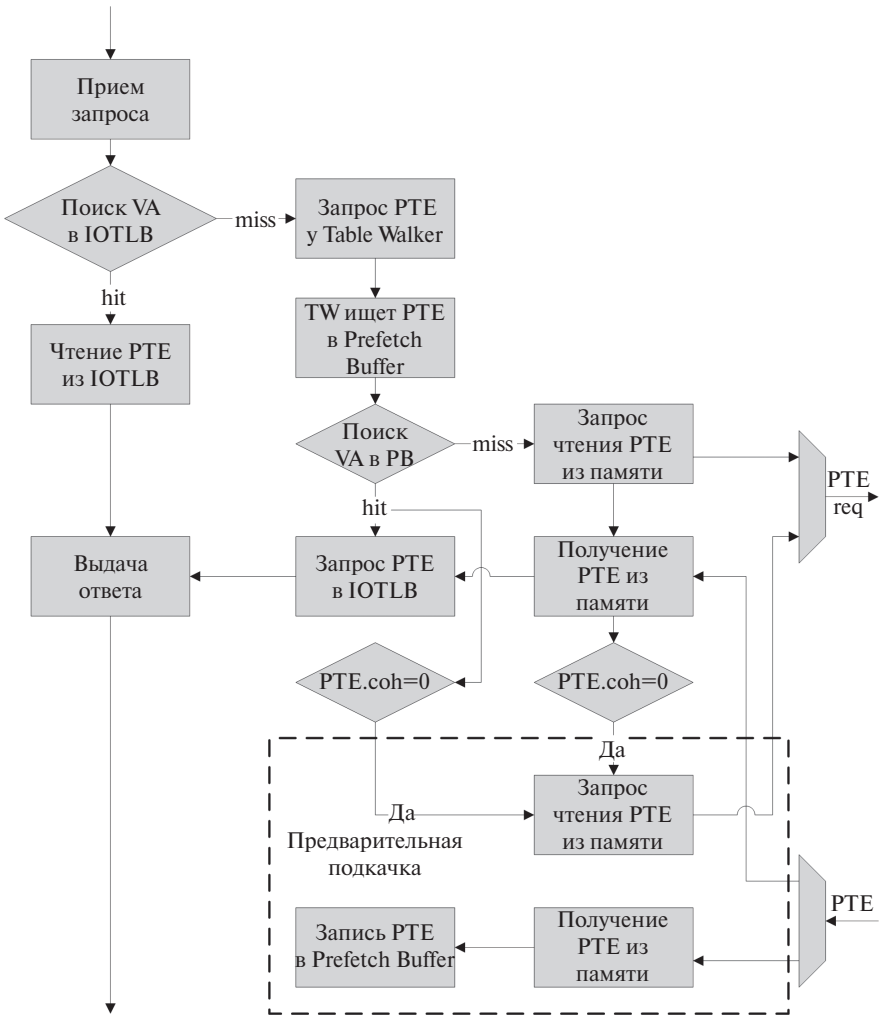

Рисунок 3. Алгоритм трансляции адреса в IOMMU МП «Эльбрус-8С2»

Таблица. Среднее время выполнения DMA-обменов

Тесты	Время в тактах		
	«Эльбрус-8С»	«Эльбрус-8С2» без подкачки	«Эльбрус-8С2» с подкачкой
Обмен небольшими блоками	13600	8300 (+63%)	8000 (+4%)
Обмен кэш-строками	1850	1140 (+62%)	860 (+33%)

Если к этому времени прибавить время чтения PTE, получим 1030 тактов на пересылку одной страницы. Таким образом, прирост производительности за счет отсутствия простоя составит чуть больше 7%. В этой связи следует отметить, что с увеличением пропускной способности канала ввода-вывода прирост производительности за счет подкачки будет расти, и при скорости 16 Гб/с он составит 23%.

В качестве экспериментальных результатов можно привести замеры времени выполнения обменов размером в несколько страниц на rtl-моделях МП «Эльбрус-8С» и «Эльбрус-8С2», к которым в качестве внешнего устройства подключен генератор DMA-обменов с пропускной способностью 16 Гб/с. Генератор копирует данные оперативной памяти, т. е. сначала исполняет DMA-чтение в свой внутренний буфер, а затем DMA-запись в оперативную память. Запускались два вида тестов: копирование массива небольшими блоками данных с размером менее одной кэш-строки и копирование массива блоками по одной кэш-строке. Общий размер всех обменов в тесте составляет четыре страницы (16 Кб). Результаты прогона приведены в таблице, где в скобках указан прирост производительности по сравнению с предыдущей конфигурацией.

Заключение

В результате введения описанных в статье усовершенствований производительность устройства трансляции виртуальных адресов в физические IOMMU в МП «Эльбрус-8С2» увеличилась более чем на 60%. Дальнейший прирост производительности планируется получить за счет совершенствования алгоритмов вытеснения, в особенности для консистентных обменов, и введения дополнительного полностью программируемого буфера TLB. Кроме того, структура нового IOMMU подготовлена к реализации многоуровневой таблицы страниц и двухуровневой трансляции для полной виртуализации внешних устройств.

СПИСОК ЛИТЕРАТУРЫ

1. AMD I/O Virtualization Technology (IOMMU) Specification Revision 2.0 [Электронный ресурс]. 2011. URL: http://developer.amd.com/wordpress/media/2012/10/48882.pdf
2. Intel Virtualization Technology for Directed I/O (VT-d) Architecture Specification [Электронный ресурс]. October 2014. URL: http://www.intel.com/content/dam/www/public/us/en/documents/product-specifications/vt-directed-io-spec.pdf
3. Ben-Yehuda M. et al. Utilizing IOMMUs for virtualization in Linux and Xen. Proceedings of the Linux Symposium, 2006. Ottawa, Ontario, Canada.
4. Ben-Yehuda M., Xenidis J., Ostrowski M. Price of Safety: Evaluating IOMMU Performance. Proceedings of the Linux Symposium 2007. Ottawa, Ontario, Canada.
5. ARM SMMU [Электронный ресурс]. URL: https://www.arm.com/products/system-ip/controllers/system-mmu.php
6. Jouppi N. P. Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers. SIGARCH Comput. Archit. News 18, 3a (1990), pp. 364–373.
7. Amit N., Ben-Yehuda M., Yassour B. A. IOMMU: Strategies for Mitigating the IOTLB Bottleneck. WIOSCA 2010 – Sixth Annual Workshop on the Interaction between Operating Systems and Computer Architecture, Saint Malo, France, Jun 2010.
8. Hennessy J., Patterson D. Computer Architecture: A Quantitative Approach. 5th ed. Morgan Kaufmann, 2011.

ИНФОРМАЦИЯ ОБ АВТОРЕ

Поляков Никита Юрьевич, инженер, АО «МЦСТ», 119334, Москва, ул. Вавилова, д. 24, тел.: 8 (499) 135-31-08, e-mail: polyakov_n@mcst.ru.

For citation: Polyakov N. Yu. DMA virtual address translation in «Elbrus-8C2» processor. Voprosy radioelektroniki, 2017, no. 3, pp. 22–26.

N. Yu. Polyakov

DMA VIRTUAL ADDRESS TRANSLATION IN «ELBRUS-8C2» PROCESSOR

Continuous increase of main memory size and security requirements growth of modern computers lead to DMA virtual address resolution support in operating systems, I/O interfaces, and processors. In this paper we consider the first implementation of

DMA virtual address translation unit in «Elbrus» architecture processor. Then we describe performance improvements of its next version, which has been implemented in «Elbrus-8C2» processor.

Keywords: IOMMU, DMA, Elbrus, virtualization, IOTLB, prefetch.

REFERENCES

1. [AMD I/O Virtualization Technology (IOMMU) Specification Revision 2.0]. Available at: http://developer.amd.com/wordpress/media/2012/10/48882.pdf
2. [Intel Virtualization Technology for Directed I/O (VT-d) Architecture Specification]. October 2014. Available at: http://www.intel.com/content/dam/www/public/us/en/documents/product-specifications/vt-directed-io-spec.pdf
3. Ben-Yehuda M. et al. Utilizing IOMMUs for virtualization in Linux and Xen. *Proceedings of the Linux Symposium,* 2006. Ottawa, Ontario, Canada.
4. Ben-Yehuda M., Xenidis J., Ostrowski M. Price of Safety: Evaluating IOMMU Performance. *Proceedings of the Linux Symposium,* 2007. Ottawa, Ontario, Canada.
5. [ARM SMMU]. Available at: https://www.arm.com/products/system-ip/controllers/system-mmu.php
6. Jouppi N. P. Improving direct-mapped cache performance by the addition of a small fully-associative cache and prefetch buffers. *SIGARCH Comput. Archit. News 18*, 3a (1990), pp. 364–373.
7. Amit N., Ben-Yehuda M., Yassour B. A. IOMMU: Strategies for Mitigating the IOTLB Bottleneck. *WIOSCA 2010 – Sixth Annual Workshop on the Interaction between Operating Systems and Computer Architecture,* Saint Malo, France, Jun 2010.
8. Hennessy J., Patterson D. Computer Architecture: A Quantitative Approach. 5th ed. Morgan Kaufmann, 2011.

AUTHOR

Polyakov Nikita, engineer, JSC «MCST», 24, Vavilova st., Moscow, 119334, Russian Federation, tel.: +7 (499) 135-31-08, e-mail: polyakov_n@mcst.ru.