Способ снижения времени задержки трафика в непрозрачных мостах коммутатора PCI Express

СУХИХ А.В.
Поволжский государственный технологический университет, Йошкар-Ола

Введение

Классификация параллельных вычислительных систем в настоящее время очень разнообразна, но особую популярность имеет отдельный класс параллельных архитектур – кластерные системы. Под кластером понимается совокупность вычислительных узлов, объединенных сетью. Правильно собранный и настроенный кластер позволяет получить преимущество в производительности, пропорциональное количеству вычислителей. Как правило, кластерные системы интенсивно используются для проведения вычислений. Предприятия и организации чаще всего приобретают кластеры для решения потока задач. Зачастую потребности желающих воспользоваться вычислительными ресурсами превосходят доступный объем ресурсов [1].

В связи с этим большую популярность приобрели системы, основанные на дублировании вычислительных устройств, которые особым образом по заданным алгоритмам могут сообща решать сложные задачи, работая параллельно. Таким образом складывается ситуация, когда даже параллельно работающий коллектив вычислителей не удовлетворяет всех потребностей. Встает вопрос повышения производительности кластера и решается он различными способами [2, 3, 4, 5].

Первый способ заключается в увеличении количества вычислителей, что ведет к значительным финансовым затратам не только на сами вычислители, а также на увеличение инфраструктуры: площади под серверные стойки, систему охлаждения и электропитания, коммутационную среду и т.д.

Второй способ – это модернизация коммутационной среды вычислительного кластера.

В современных кластерах на первый план выходит увеличение производительности коммутационной среды, которая отвечает за обмен данными между узлами обработки данных. Чем выше скорость обмена и меньше вносимая коммутационной средой задержка, тем выше общая производительность кластера. Для решения этой проблемы разработаны специальные аппаратные структуры и протоколы обмена, которые легли в основу таких стандартов как InfiniBand или Fibre Channel. Наряду с ними, популярным решением для объединения вычислителей в кластер является протокол PCI Express (PCIe).

Цель работы

Снижение времени задержки трафика в непрозрачных портах коммутатора PCI Express, соединяющего разные адресные домены, за счёт модификации алгоритма трансляции адресов.

Решаемые задачи

Формализация алгоритма трансляции адреса получателя, извлекаемого из пакетов PCI Express.

Модификация алгоритма для достижения цели работы, т.е. повышение производительности непрозрачного моста.

Базовый алгоритм трансляции адреса получателя PCIe

Обмен информацией между узлами разных адресных доменов возможен за счёт использования механизма трансляции адресов. Трансляция адресов выполняется по особым алгоритмам, которые вносят задержку в прохождение пакета по коммутационной

системе, поэтому модификация существующих алгоритмов трансляции позволит снизить эти издержки и повысить общую производительность всей кластерной системы.

Для формализации и исследования существующих алгоритмов трансляции, которые лежат в основе работы современных интегральных микросхем, рассмотрим устройство PES32NT24G2s от фирмы «Integrated Device Technology», которая является одной из лидирующих мировых компаний, выпускающих микрочипы, работающие с протоколом PCI Express. Устройство представляет собой коммутационный элемент с конфигурируемым набором прозрачных и непрозрачных мостов [6].

В настоящее время существующий алгоритм трансляции адреса в НМ предполагает использование специального поля «Index» в пакете PCI Express, по значению которого происходит поиск транслированных адресов в регистрах НМ. Данное поле не содержится в классическом формате пакета PCI Express, который описан в «PCI Express Base 3.0 Specification». Следовательно, для использования этого поля требуется внести изменения в протокол передачи PCI Express, тем самым усложняя реализацию способа трансляции и добавляя дополнительные функции по формированию пакета в оконечное устройство PCI Express. Для описания алгоритмов были использованы обозначения регистров НМ (см. рис. 1), а также следующие принятые формализации:

- **RgBst[i] -** i-ый бит регистра BarSetup непрозрачного моста;
- **RgTBA[i]** - i-ый регистр транслированного базового адреса (для прямой трансляции адресов), соответствует i-му регистру BAR[i].

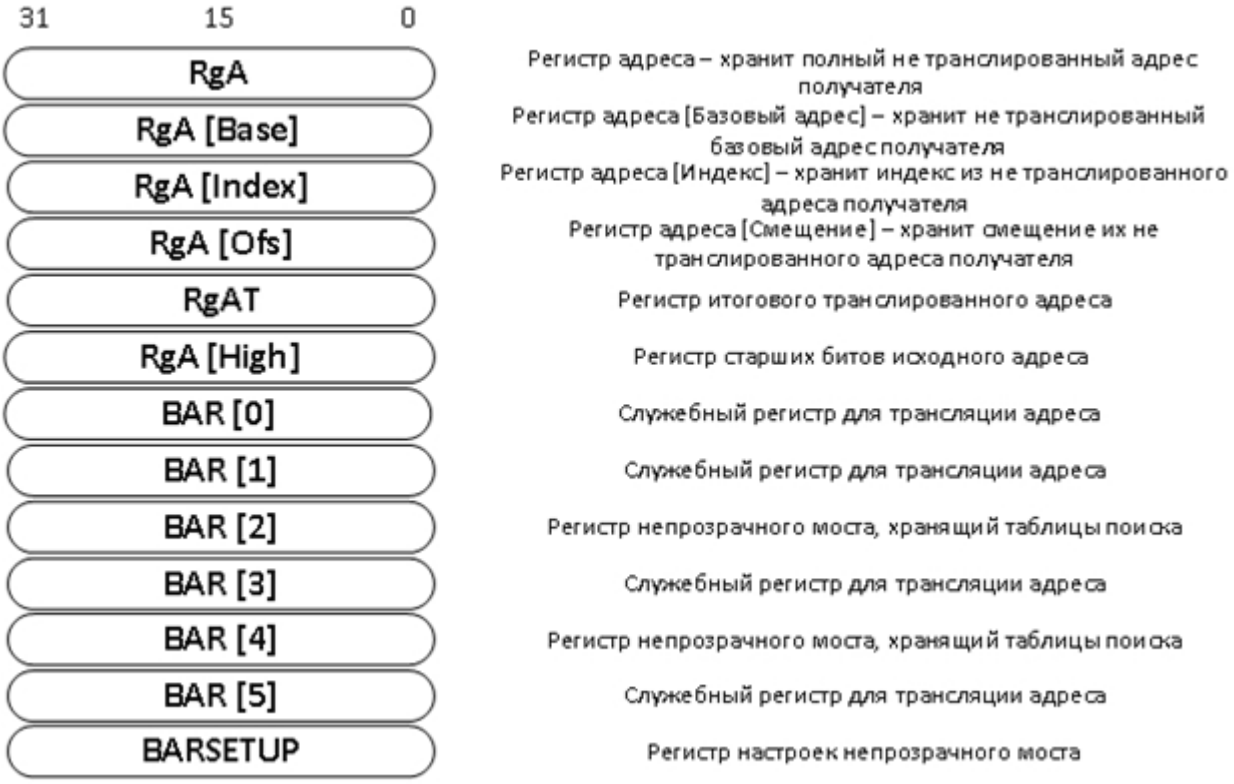

Рис. 1. Регистры непрозрачного моста

В качестве иллюстрации алгоритма трансляции адреса рассмотрим случай, когда требуется передать пакет от хоста А хосту В по кабельной системе PCI Express с использованием внешнего коммутатора с трансляцией адресов (рис. 2).

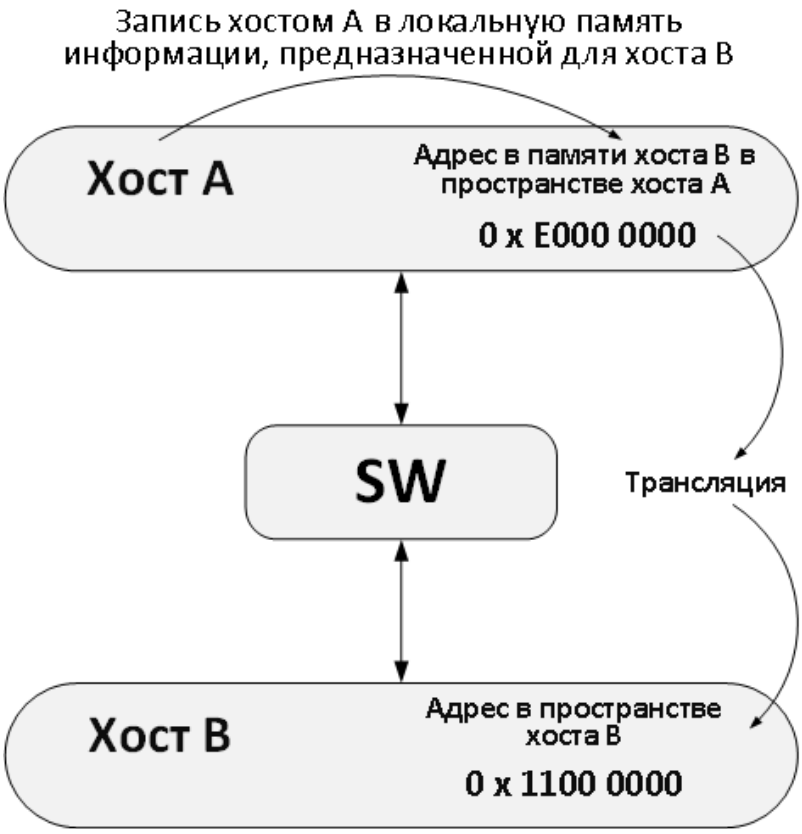

Рис. 2. Пример транзакции на шине PCI Express с использованием внешнего коммутатора (SW)

Схема трансляции адреса в НМ для этого случая приведена на рисунке 3.

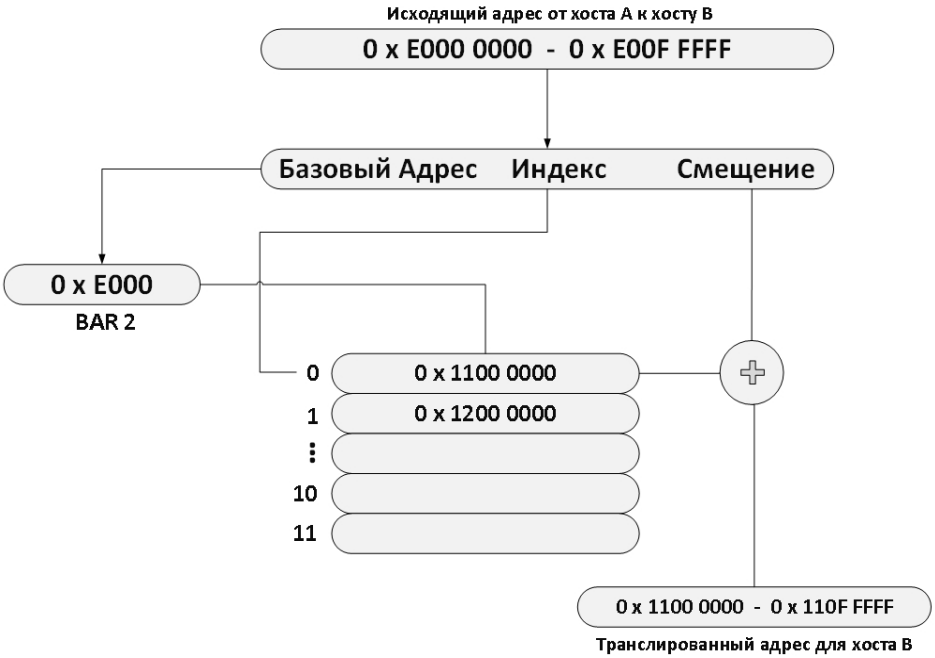

Рис. 3. Схема трансляции адреса в непрозрачном мосту PCI Express

Исходный алгоритм трансляции адреса в НМ представлен на рис.4.

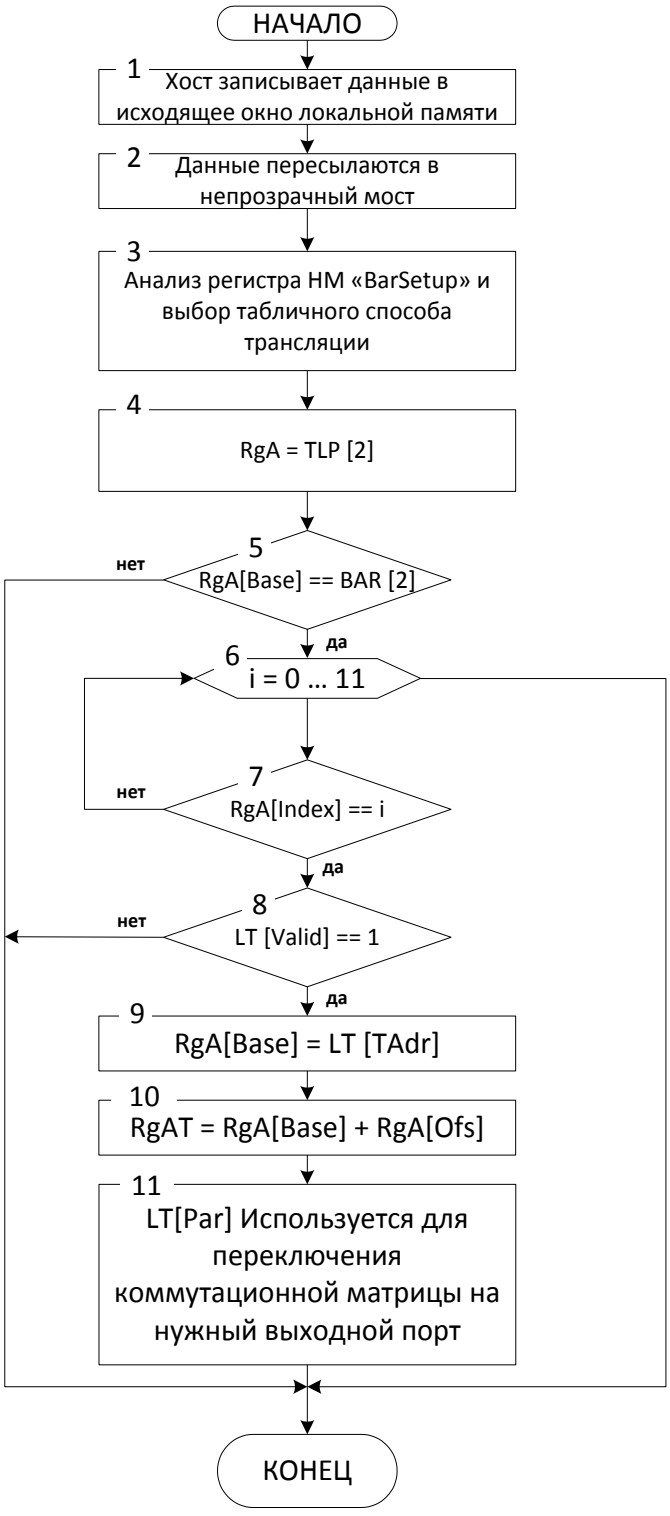

Рис. 4. Алгоритм трансляции адреса в устройстве PES32NT24G2s и его аналогах

В начале транзакции устройство-передатчик записывает данные, которые следует передать, в исходящее окно своей локальной памяти **(1)**. Исходящее окно представляет собой диапазон адресного пространства хоста, которое формируется в каждом устройстве, подключенном к коммутатору PCI Express, в момент инициализации и служит для записи в него данных, предназначенных для передачи другому узлу. На следующем шаге **(2)** сформированный пакет протокола пересылается в НМ коммутатора. Так в настоящем патенте рассматривается только табличный способ трансляции адресов, регистр НМ «BarSetup» заведомо сконфигурирован на применение этого способа **(3)**. Затем второе

двойное слово пакета транспортного уровня протокола PCI Express (TLP), содержащее в себе транслируемый адрес, записывается в адресный регистр НМ RgA **(4)**. На шаге **(5)** базовый адрес транслируемого адреса, содержащийся в регистре НМ RgA[Base] сравнивается с базовым адресом, содержащимся в регистре НМ BAR[2]. Если адреса совпадают, то на шаге **(6)** происходит поиск записей по индексу внутри таблицы трансляции адресов. Значение индекса выбирается из адресной части пакета TLP. Всего в таблице может находиться до 12 записей. В случае успешного поиска на шаге **(7)** и проверки поля действительности **(8)** (LT[Valid]), что подтверждает актуальность записи в таблице поиска, базовый адрес, содержащийся в RgA[Base] заменяется на базовый адрес соответствующей записи в таблице поиска (LT[TAdr]), иными словами происходит трансляция адреса **(9)**. Затем в регистр транслированного адреса (RgAT) записывается транслированный базовый адрес из LT[TAdr] и прибавляется смещение из исходного адреса, хранящееся в регистре RgA[Ofs] **(10)**. В конце параметр таблицы поиска LT[Par], определяющий номер порта, на который следует передать пакет с транслированным адресом, используется для определения номера выходного порта.

Формат таблицы трансляции адресов НМ приведен в табл. 1.

Таблица 1

Формат таблицы трансляции адресов

Индекс	**Транслированный адрес**	**Валидность**
0	0x1100 0000	1
1	0x1600 0000	1
2	0x1D00 0000	1
3	0xB100 C000	0

Модификация алгоритма

Предлагаемый автором способ позволяет отказаться от использования специального поля «Index» и тем самым избавиться от необходимости внесения изменений в протокол PCI Express.

Суть предлагаемого решения заключается в следующем. Поскольку так или иначе транслированный и исходный адреса однозначно сопоставляются друг другу, то целесообразней не проводить поиск в таблице трансляции по искусственно введенному полю «Index», а по старшим битам исходного адреса определять соответствующий транслированный адрес. Таким образом, формат таблицы трансляции изменится. Он приведен в табл. 2.

Таблица 2

Формат модифицированной таблицы трансляции адресов

Старшие биты исходного адреса	**Транслированный адрес**	**Валидность**
0x0100	0x1100 0000	1
0x0200	0x1600 0000	1
0x0400	0x1D00 0000	1
0x0500	0xB100 C000	0

Соответственно, алгоритм поиска также претерпит некоторые изменения. Аппаратная реализация устройства коммутатора не изменится, поскольку в новом алгоритме поиска будут задействованы те же самые наборы регистров. Измененный алгоритм представлен на рис. 5., а модифицированная схема трансляции адресов на рис. 6.

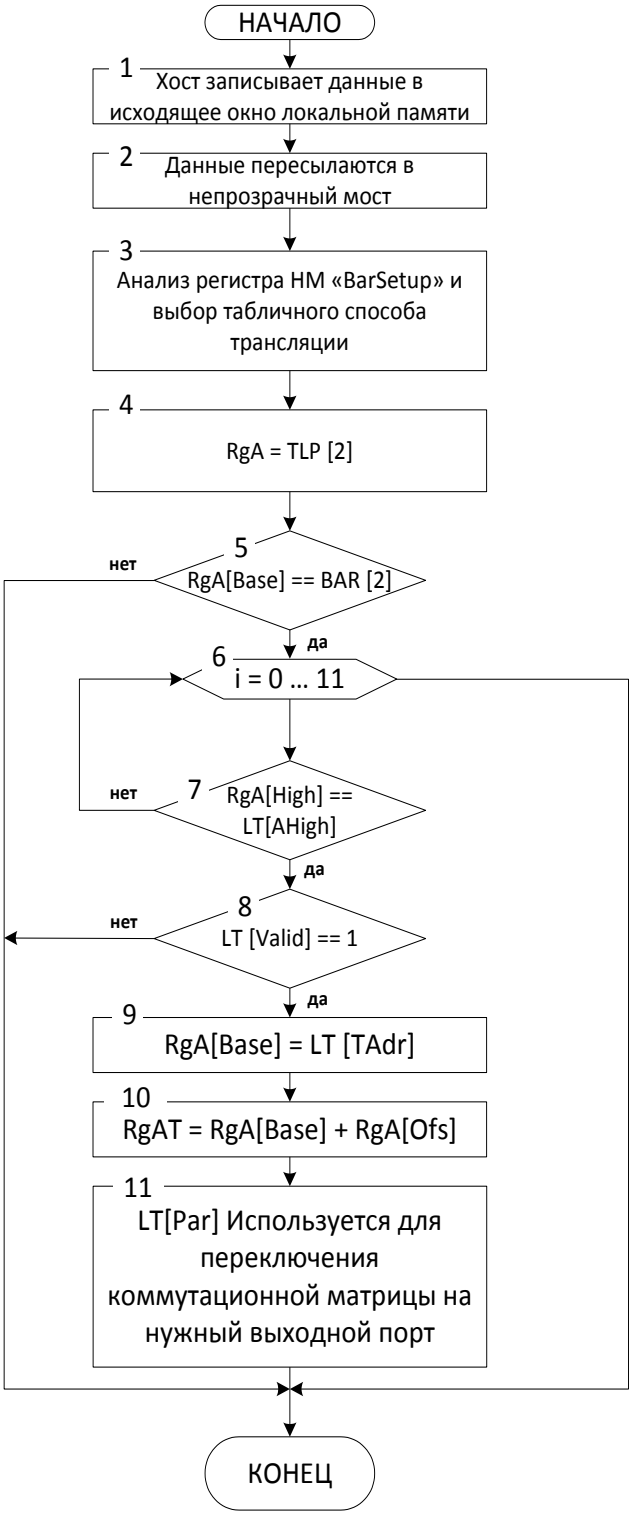

Рис. 5. Модифицированный алгоритм трансляции адреса в непрозрачном порту PCI Express

Ключевой особенностью данного алгоритма является то, что на шаге **(7)** старшие биты исходного адреса (RgA[High]) сравниваются со значением (LT[AHigh]) в таблице поиска.

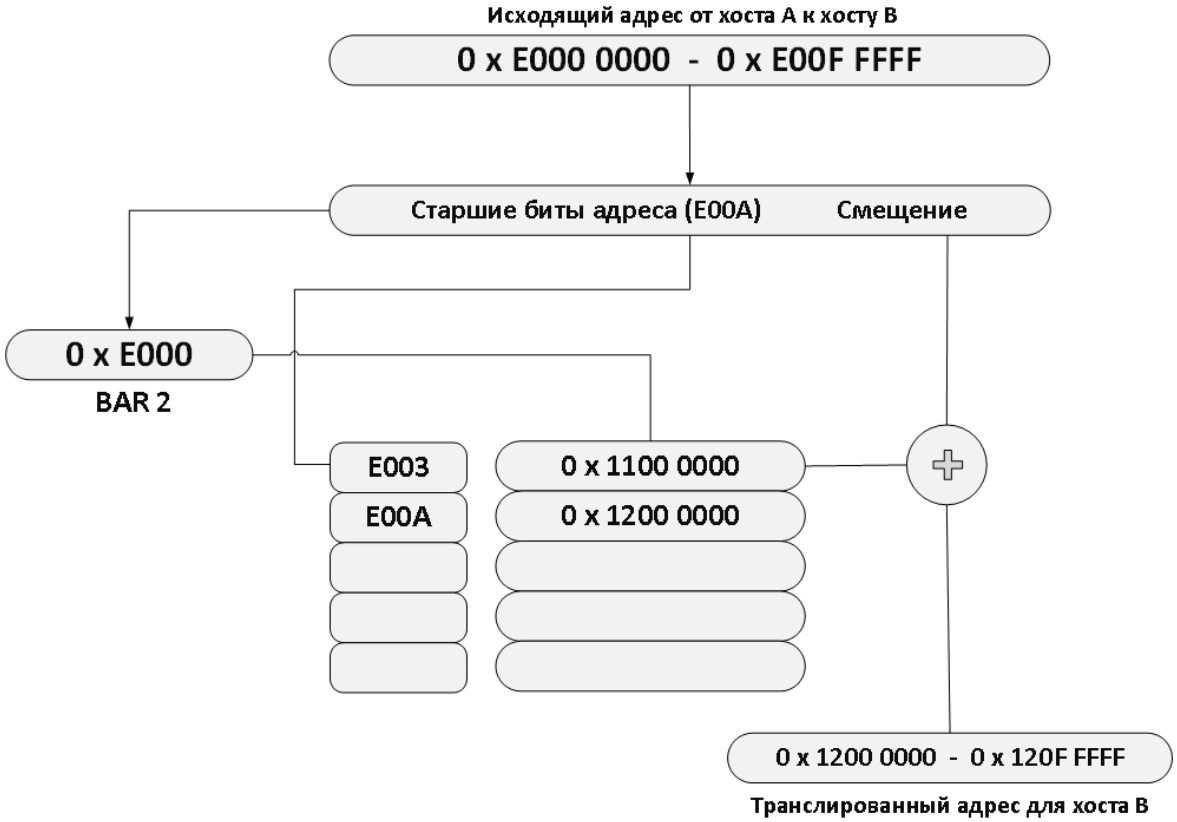

Рис. 6. Модифицированная схема трансляции адреса в непрозрачном мосту PCI Express

Выводы

Для определения алгоритмической сложности базового и предложенного алгоритмов, а также для оценки полученного выигрыша в работе проводился расчёт трудоемкости алгоритмов в элементарных машинных операциях (элОп) марковским методом [7].

По результатам расчетов марковским методом для **базового** алгоритма были получены следующие значения трудоемкости:

- $Q_{1\,avg}$ = 208,935 элОп;
- $Q_{1\,min}$ = 49,107 элОп;
- $Q_{1\,max}$ = 300,591 элОп.

Для **модифицированного** алгоритма:

- $Q_{2\,avg}$ = 117,621 элОп;
- $Q_{2\,min}$ = 37,707 элОп;
- $Q_{2\,max}$ = 208,935 элОп.

Таким образом, можно сделать вывод о снижении **средней** трудоемкости операции трансляции адреса в 1,77 раза, **минимальной** трудоемкости в 1,3 раза и **максимальной** в 1,43 раза.

Снижение трудоемкости вычисления транслированного адреса сокращает время его расчета процессором. Например, время вычисления адреса процессором, который построен на ARM-архитектуре Cortex-A9 и имеет производительность до 4 GFLOPS [8], для средней трудоемкости составит 29 нс (вместо 52 нс для исходного алгоритма), для минимальной трудоемкости – 9 нс (вместо 11 нс), для максимальной трудоемкости – 52 нс (вместо 75 нс).

Список использованной литературы

1. Корняков, К.В. Технологии построения и использования кластерных систем./ К.В. Корняков, А.В. Шишков. / НГУ им. Лобачевского, Нижний Новгород, 2007.
2. Довгаль, В.М. Повышение производительности кластеров рабочих станций с использованием веерного распределения дополнительных заданий на простаивающее оборудование. / В.М. Довгаль., С.Г. Спирин / Ученые записки. Курский государственный университет. 2012. №4 (24) ч.2.
3. Воеводин, В. В. Параллельные вычисления. / В.В. Воеводин, Вл. В. Воеводин / СПб.: БХВ-Петербург, 2002. 608 с.
4. Воеводин, Вл. В. Суперкомпьютеры и парадоксы неэффективности / Открытые системы. 2009. № 10.– С. 37–45
5. Increase of productivity of the PCI Express switch for cluster systems/ Vasjaeva N.S., Ivanov K.V., Suhih A.V.// В мире научных открытий.- Красноярск: Научно-инновационный центр, 2014. № 10(58) (Естественные и технические науки). –с. 248-262.
6. Kwok Kong. Non-transparent Bridging with IDT 89HPES32NT24G2 PCI Express® NTB Switch [Электронный ресурс] / www.idt.com : официальный сайт компании IDT. URL: https://www.idt.com/document/apn/724-non-transparent-bridging-idt-pes32nt24g2-pcie-switch
7. Ларионов, А.М. Вычислительные комплексы, системы и сети: Учебник / А.М. Ларионов, С.А. Майоров, Г.И. Новиков. – Л.: Энергоиздат, 1978. – 383с.
8. CPU FLOPS [Электронный ресурс] / http://dench.flatlib.jp/opengl/cpuflops