УДК 004.255

Виртуализация устройств прямого доступа к памяти

А.А. Аракелов, С.И. Аряшев, Р.Ш. Кабиров

Учреждение Российской академии наук Научно-исследовательский институт

системных исследований РАН, arakelov@cs.niisi.ras.ru

***Аннотация* — Описан метод виртуализации устройств прямого доступа к памяти в микропроцессоре, позволяющий отображать адреса свыше 4 ГБ для устаревших устройств, увеличить безопасность доступа устройств к памяти, предоставить гостевым операционным системам в виртуальных машинах прямой доступ к аппаратным ресурсам.**

***Ключевые слова* — виртуализация.**

I. Виртуальная память

Виртуализация в компьютерных технологиях – это абстракция вычислительных ресурсов, сокрытие настоящей аппаратной реализации и предоставление пользователю более удобной для использования системы, скрывающей действительное исполнение объекта. Виртуализация памяти процессора имеет аппаратную поддержку на всех современных процессорах: блок управления памятью обеспечивает трансляцию виртуальных адресов исполняемого кода в физические и защиту множества адресных пространств памяти.

В настоящее время виртуальная память часто организуется страничной адресацией: память делится на области фиксированной длины (страницы), которые являются минимальной единицей выделяемой памяти. Процессор содержит в себе небольшой объём сверхбыстрой ассоциативной памяти, т.н. TLB (Translation Lookaside Buffer), в котором содержится преобразование нескольких (часто 64) виртуальных адресов страниц в физические. Все обращения процессора к памяти подлежат трансляции адресов через TLB. Процесс обращается к памяти с помощью адреса виртуальной памяти, который содержит в себе номер страницы и смещение внутри страницы.

Так как 64 строк таблицы явно недостаточно для реальных задач, в архитектуре используются таблицы страниц, размещённые в основной памяти. Каждая таблица страниц содержит в себе массив входов таблицы страниц; вход таблицы содержит в себе физический адрес и флаги (страница отображена, страница доступна только на чтение, страница доступна из режима пользователя, страница не доступна на исполнение и пр.). Так как число входов в одной таблице ограничено размером входа и размером страницы, используется многоуровневая организация таблиц, часто 2 или 3 уровня, иногда 4 уровня (для 64-х разрядных архитектур). В случае двух уровней используется "директория" страниц, имеющая в себе входы, указывающие на физические адреса таблиц страниц. Старшие биты виртуального адреса понимаются как номер входа в директорию, средние – как номер входа в таблицу, младшие (адрес внутри страницы) попадают в физический адрес без трансляции. Адрес размещения таблицы первого уровня в памяти программно записывается в специальном регистре процессора (напр. регистр Context для архитектуры MIPS), в случае трехуровневой таблицы таблица второго уровня индексируется значением, найденным в таблице первого уровня, см. рис. 1.

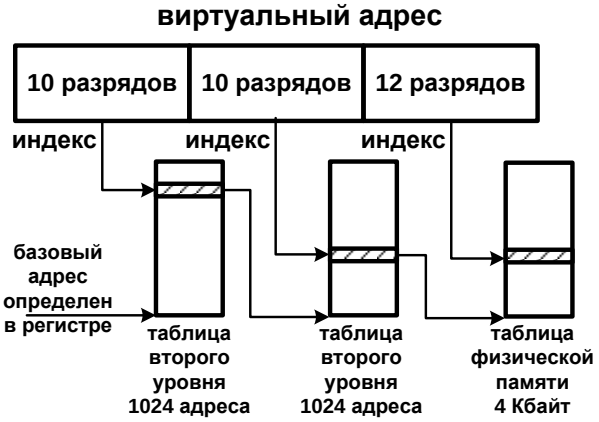

Рис. 1. Трехуровневая трансляция адреса

Содержание ячейки TLB на примере архитектуры MIPS64 показано на рис. 2. Минимальный размер страницы – 1 Кбайт (версия 2 спецификации), маска позволяет использовать страницы размером до 256 Мбайт. Поле R определяет регион памяти. Каждой виртуальной странице сопоставлено две физических. Флаги (V - физическая страница действительна, D – страница защищена от записи, C – политика кэширования для страницы) определяют трансляцию для данного виртуального адреса.

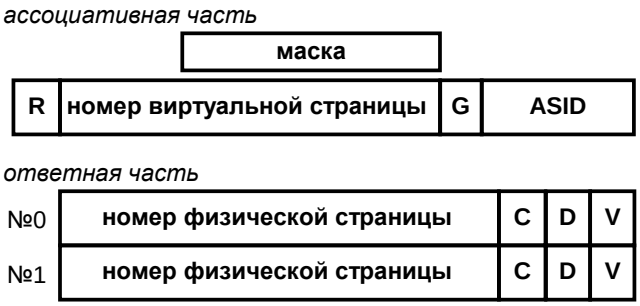

Рис. 2. Ячейка TLB

II. Виртуализация ввода/вывода

Устройства прямого доступа к памяти (ПДП) в системе-на-кристалле (СнК) обычно имеют стандартные интерфейсы для подключения. Шина PCI использует минимум 32 разряда для адреса, и позволяет использовать, возможно, всё адресное пространство в 32-х разрядной системе (но не в 64-разрядной); а, например, шина ISA имеет 24 адресных линии, и позволяет адресовать только 16 МБ памяти. Устройства с 32-х разрядной адресацией могут использовать только 4 ГБ памяти, в то время, как 64-х разрядные системы способны предоставить большее количество памяти. При проектировании устройств зачастую происходит повторное использование IP-блоков, как с целью сокращения ресурсов на разработку, так и для сохранения совместимости программно-аппаратного комплекса; например, для сохранения всех существующих драйверов устройств. Так, например, при создании нового 64-х разрядного процессора возможно использование IP-блока Ethernet контроллера 10/100 Мбит, подключенного по внутренней 32-х разрядной шине. Контроллер Ethernet в этом случае не способен обратиться в произвольную ячейку памяти. При этом операционная система (ОС) должна теперь разделять нижние адреса памяти и верхние адреса памяти, не доступные некоторым устройствам, и следить, чтобы работа с такими устройствами велась только с адресами нижних областей памяти. Чтобы произвести передачу из такого устройства в область верхних адресов памяти обычно ОС выделяет область памяти в нижних адресах, доступных устройству, т.н. "буфер отскока", производит передачу в этот буфер, а потом копирует результат в требуемую область в верхних адресах памяти. Такой способ значительно снижает производительность быстродействующих устройств ввода/вывода.

В ОС со страничной виртуальной памятью, таких, как Windows и семейство UNIX, непрерывный регион виртуальных адресов может быть реализован разрывно расположенными физическими страницами. Доступ к такому региону для устройства ПДП представляет собой довольно сложную задачу.

Решение этой задачи требует выявления физических страниц, реализующих регион. Далее становится возможным нахождение физических адресов страниц региона, которые в общем случае не являются непрерывными и формируют так называемый scatter-gather list ("список рассеяния/сборки") – SGL.

Задача исполнения ПДП по таковому списку может быть решена одним из следующих способов:

1) использование "буфера отскока". Такой метод довольно прост в реализации, однако требует значительных затрат времени и ресурсов памяти (дополнительное место в памяти под буфер отскока);

2) разбиение операции на подоперации по границам элементов SGL, с прерыванием в конце каждой операции. Этот метод подразумевает обработку большого количества прерываний от устройств;

3) поддержка SGL самим устройством, с требованием размещения SGL, преобразованного в формат, специфичный для устройства, в физически непрерывном регионе основной памяти. Устройство читает SGL тем же механизмом ПДП с захватом шины, что и собственно данные, тем самым реализуя функциональность некоего процессора, читающего и исполняющего свою собственную "программу", реализованную как список дескрипторов SGL. Этот подход требует высокой сложности самого устройства ПДП.

4) поддержка SGL в межшинном оборудовании, при которой представление физически разрывного буфера со стороны устройства выглядит физически непрерывным, см. рис. 3. Такое оборудование называется IOMMU (англ. IO memory management unit – блок управления памятью устройств ввода/вывода). IOMMU требует высокой сложности аппаратуры, уже на уровне платформы.

Блок управления памятью устройств ПДП в микропроцессоре отвечает за управление доступом к памяти каких-либо устройств помимо процессора. Аналогично блоку управления памятью процессора, IOMMU обеспечивает трансляцию виртуальных адресов устройств ПДП в физические адреса и защиту адресных пространств. Повторное использование IP-блока трансляции позволяет сократить затраты на разработку сложной логики такого блока.

Использование IOMMU позволяет не только разрешить проблему адресации больших объемов памяти и расположенных нелинейно страниц данных, но также обеспечивает дополнительную защиту памяти и поддержку гостевых ОС, работающих в виртуальных машинах. К недостаткам использования IOMMU можно отнести некоторое снижение производительности, возникающее за счет управления и обслуживания блока и собственно процесса трансляции адресов, а также потребление физической памяти, занятой под таблицы трансляции. Виртуализация устройств ПДП обеспечивает защиту памяти следующим образом: устройство

не может получить доступ к областям памяти, которые не были явно отображены в IOMMU для этого устройства перед началом работы. Помимо этого, IOMMU может запретить запись в какие-либо отображенные области.

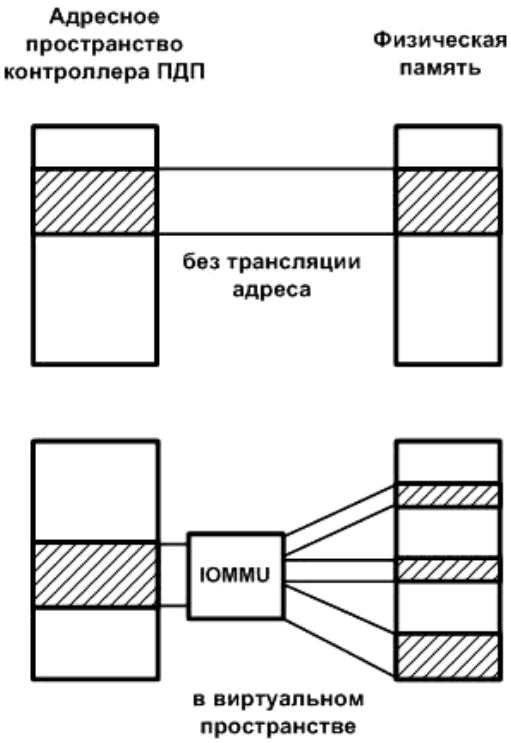

Рис. 3. Трансляция адресов контроллера ПДП

В случае ОС, работающей в виртуальной машине, гипервизор или главная ОС конфигурирует аппаратуру для работы другой ОС, а затем передает управление последней. Гостевая ОС обычно не имеет информации о физических адресах в машине. При этом если гостевая ОС получит прямой доступ к устройствам ПДП, то использование виртуальных адресов при работе с физической памятью приведет к разрушению информации в памяти. Этого можно избежать, если гипервизор или главная ОС будет перехватывать запросы к устройствам и транслировать адреса, однако такой способ приводит большим задержкам при работе. IOMMU решает эту проблему отображением адресов для устройств ПДП в соответствии с таблицами трансляции, используемыми самой гостевой ОС, т.е. гостевая ОС получает прямой доступ к устройствам ПДП без вмешательства в работу драйверов.

III. Реализация

Для работы с виртуальной памятью процессор имеет в составе блок управления памятью, транслирующий виртуальные адреса исполняемого кода в физические адреса, и обеспечивающий защиту доступа к адресным множествам. В целях сокращения времени разработки данный блок, практически являясь спроектированным и отлаженным IP-блоком, может быть повторно использован для виртуализации памяти устройств ПДП. При этом возможно несколько архитектурных решений по реализации блока в системе-на-кристалле:

1) использовать отдельный блок IOMMU для каждого канала от устройств ПДП, где требуется трансляция адресов;

2) использовать один блок IOMMU для всех устройств ПДП, разместив его в контроллере памяти системного контроллера, см. рис. 4

Поскольку обращение в память в момент времени может производить только одно устройство, такое размещение возможно. Первый случай представляется оптимальным, но в процессе реализации потребовал значительной площади кристалла для блоков трансляции, что привело к отказу от такого решения.

Оценка второго подхода была произведена на прототипе разрабатываемого в НИИСИ процессора с архитектурой MIPS64. Единый для всех устройств ПДП блок IOMMU, встроен в системный контроллер, при этом возможно использовать только 64 ячейки трансляции для всех устройств ПДП. Трансляция адресов вносит дополнительную задержку в тракте контроллера памяти. В зависимости от устройства ПДП, обращающегося в память, эта задержка приводит к уменьшению производительности от ~0.8% для контроллера RapidIO до ~6,5% для контроллера DMA к кэш-памяти второго уровня.

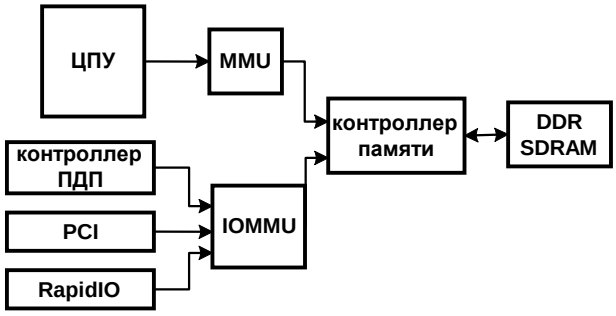

Рис. 4. Расположение IOMMU в системном контроллере

Конфигурация используемого блока управления памятью осуществляется программно в привилегированном режиме. В случае возникновения исключительной ситуации процессору выставляется прерывание. При трансляции адресов устройств ПДП возможны следующие исключительные ситуации:

1) промах – запрошенный виртуальный адрес не найден в таблице трансляции;

2) недействительная ячейка – запрошенный виртуальный адрес найден, но помечен, как недействительный;

3) защита от записи – устройство требует запись по виртуальному адресу, которой отображен в адресное пространство, защищенное от записи.

Обработка исключительных ситуаций может быть выполнена как программно центральным процессором, так и аппаратно. Например, при отсутствии виртуального адреса в буфере трансляции, IOMMU может аппаратно обратиться к таблице страниц трансляции в памяти, заполнить ячейку, и продолжить работу. Если аппаратное обращение к таблице страниц трансляции не реализовано, а также для остальных типов исключительных ситуаций, то требуется протоколировать ошибки. Высокоскоростные устройства ПДП могут вызывать исключительные ситуации с высокой интенсивностью; журнал ошибок, содержащий виртуальные адреса, вызвавшие ошибки трансляции, обычно хранится в памяти по адресу, указанному в регистре управления IOMMU. Также ведется подсчет исключительных ситуаций.

Поскольку потоковое устройство ПДП не всегда можно моментально остановить, в системном контроллере требуется предусмотреть механизм, предупреждающий разрушение данных при ошибочных ситуациях, например, маскирование данных в момент записи по "плохому" физическому адресу. Для повышения безопасности также необходимо предусмотреть защиту от чтения из неотображенных адресных пространств. Другим аспектом высокоскоростных устройств ввода/вывода являются ошибочные ситуации

Для IOMMU требуется набор регистров для управления, аналогичный регистрам MMU процессора, и расположенный в области адресов, доступных для ОС только в привилегированном режиме. Для конфигурации IOMMU можно не вводить новые инструкции для процессора, а работать через дополнительные управляющие регистры, расположенные в пространстве памяти, таким образом не изменяя архитектуру процессора. TLB процессора производит трансляцию виртуальных адресов в комбинации "виртуальный адрес + идентификатор процесса". Перед передачей управления пользовательскому процессу ОС конфигурирует блок трансляции для этого процесса и записывает идентификатор в специальный регистр процессора ASID. Для IOMMU такой механизм можно оптимизировать следующим образом: определить для каждого устройства ПДП специальный регистр IO_ASID, тогда одни и те же виртуальные адреса могут отображаться в разные физические адреса для разных устройств. Предположим, система-на-кристалле содержит процессор, а также набор устройств ввода/вывода: IP-блок 32-х разрядный мост PCI-PCI, IP-блок контроллера Ethernet 10/100, подключаемый через PCI32, контроллер RapidIO, подключаемый через 64-х разрядную шину AMBA AXI, контроллер ПДП для копирования из кэш-памяти 2го уровня процессора во внешнюю память, подключенный по внутренней 64-х разрядной шине. Каждое устройство ПДП будет иметь свой регистр IO_ASID, такая система показана на рис. 5. Для мостов PCI/PCIe™/PCI-X® возможно более точное различение подключенных устройств при формировании идентификатора как комбинации конфигурационных параметров {шина, устройство, функция}.

IV. Заключение

Использование блока управления памятью для устройств ввода/вывода позволяет:

1) использовать всю доступную физическую память для устройств, изначально не имеющих такую возможность, без использования буферов отскока и копирования больших объемов памяти;

2) обеспечить дополнительную защиту адресных пространств от несанкционированного доступа и некорректно работающих устройств ввода/вывода;

3) предоставить ОС, работающим в виртуальных машинах, прямой доступ к аппаратуре.

Повторное использование разработанного для процессора и отлаженного блока трансляции позволяет сократить цикл разработки СнК.

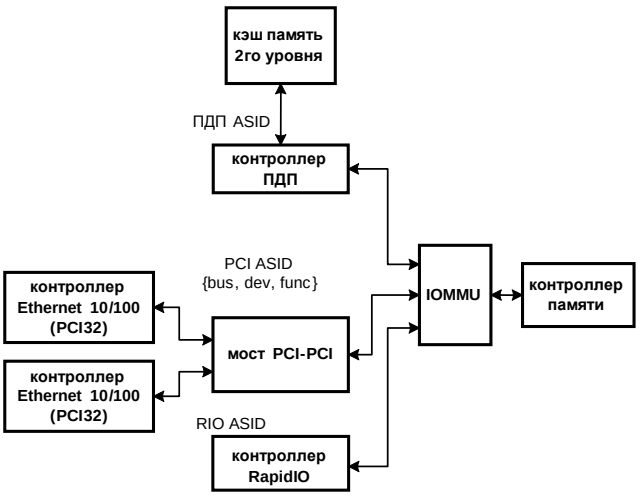

Рис. 5. Использование идентификаторов процесса в IOMMU

Литература

[1] Jason R. Thorpe, A Machine-Independent DMA Framework for NetBSD // Proceedings of the annual conference on USENIX Annual Technical Conference. - 1998. - P. 30.

[2] Bruce Jacob , Trevor Mudge, Virtual Memory: Issues of Implementation // Computer. – 1998. - V. 31. - N. 6. - P. 33-43.

[3] Drebes, R.J., Nanya, T., Limitations of the Linux Fault Injection Framework to Test Direct Memory Access Address Errors // Dependable Computing, 2008. PRDC '08. 14th IEEE Pacific Rim International Symposium. - Dec 2008. - P. 146-152.