Для цитирования: Деменко Р. В., Трофимов В. Б. Аппаратная поддержка виртуализации системы прерываний в микропроцессорах семейства «Эльбрус» // Вопросы радиоэлектроники. 2018. № 2. С. 40–44.
УДК 004.318

Р. В. Деменко[1, 2], В. Б. Трофимов[1]

[1] АО «МЦСТ», [2] МФТИ (ГУ)

АППАРАТНАЯ ПОДДЕРЖКА ВИРТУАЛИЗАЦИИ СИСТЕМЫ ПРЕРЫВАНИЙ В МИКРОПРОЦЕССОРАХ СЕМЕЙСТВА «ЭЛЬБРУС»

В современных многоядерных микропроцессорах реализуется архитектурная поддержка виртуализации аппаратных ресурсов, с целью уменьшения накладных расходов. В отличие от процессорного ядра успешные реализации аппаратной поддержки виртуализации компонентов ввода-вывода появились относительно недавно. Одним из механизмов, для которых аппаратная поддержка виртуализации целесообразна, является доставка гостевого прерывания целевому виртуальному ядру без привлечения гипервизора. В статье представлен обзор архитектуры распределенного контроллера прерываний микропроцессора «Эльбрус», а также приведены основные принципы реализации аппаратной поддержки системы прерываний. Предложено ввести гостевой набор управляющих регистров контроллера прерываний, приведен алгоритм доставки гостевых прерываний с использованием аппаратной таблицы соответствия виртуальных и физических ядер. Описаны механизмы, обеспечивающие корректность работы рассматриваемого подхода к реализации аппаратной поддержки виртуализации системы прерываний в рамках четырехпроцессорной системы.

Ключевые слова: *«Эльбрус», виртуализация, гипервизор, контроллер прерываний, виртуальные прерывания.*

Введение

Развитие технологических процессов позволяет интегрировать на одном кристалле все более сложные системы. Совершенствуется и подсистема ввода-вывода микропроцессоров (МП), в частности, аппаратура доставки прерываний. Реализуется аппаратная поддержка виртуализации системы прерываний [1–3].

В данной работе изложены основные принципы реализации аппаратной поддержки виртуализации системы прерываний для МП семейства «Эльбрус», а также сделан обзор архитектуры распределенного контроллера прерываний в МП «Эльбрус-12С», на функциональность которого опираются предлагаемые решения.

Архитектура системы прерываний микропроцессоров семейства «Эльбрус»

Внешние устройства взаимодействуют с ядром через прерывания. Прерывание характеризуется числом (вектором прерывания), которому соответствует позиция таблицы дескрипторов прерываний, хранящей указатели на программу – обработчик прерывания. Каждое ядро в многоядерной системе имеет собственную таблицу дескрипторов прерываний.

Сформированное внешним устройством прерывание доставляется в контроллер прерываний, который формирует сигнал в аппаратуру ядра о наличии необработанного прерывания. Затем системное ПО считывает вектор прерывания и вызывает соответствующую программу-обработчик.

Системное ПО взаимодействует с контроллером прерываний через управляющие регистры, отображенные на адресное пространство ввода-вывода MMIO (Memory Mapped Input Output): определение вектора полученного прерывания для начала обработки и завершение обработки прерывания осуществляются через запросы чтения и записи по этим адресам. Каждому процессорному ядру соответствует локальная часть контроллера прерываний CEPIC (Core Elbrus Programmable Interrupt Controller), содержащая соответствующие этому ядру управляющие регистры.

Прерывания разделены на четыре класса приоритетов, приоритет прерывания определяется двумя старшими разрядами вектора. Наиболее приоритетное из необработанных прерываний размещается на регистре текущего прерывания CIR (Current Interrupt Register) или, если CIR занят, фиксируется на регистре отложенных прерываний PMIRR (Pending Maskable Interrupt Request Registers). Сигнал в аппаратуру ядра о наличии прерывания выставляется, если текущий приоритет ядра CPR (Core Priority Register) меньше приоритета прерывания на CIR. В противном случае сигнал

о наличии прерывания не выставляется в ожидании снижения приоритета ядра.

Обработка прерывания начинается с чтения вектора прерывания и приоритета ядра (на момент прерывания). CIR при этом освобождается: если в процессе обработки прерывания приходит новое, более приоритетное, оно будет обработано до того, как завершится обработка первого (вложенные прерывания). Завершается обработка записью в регистр окончания обработки прерывания EOI (end-of-interrupt), при этом восстанавливается значение приоритета ядра (для корректного порядка завершения обработки вложенных прерываний).

Описанная функциональность является общепринятой, она реализована в той или иной степени у всех производителей микропроцессоров, изменения касаются:

- числа доступных векторов (так, для x86_64 доступны 256 векторов, для Elbrus – 1024 вектора);
- методов доступа к регистрам контроллера прерываний (так, в x2APIC (Intel) регистры доступны через интерфейс моделезависимых регистров MSR (Model Specific Register interface));
- структуры регистров, хранящих прерывания (так, в x2APIC не используется механизм восстановления приоритета ядра при EOI. Для корректного порядка завершения вложенных прерываний используется регистр ISR (in-service register), который явно хранит векторы всех находящихся в обработке прерываний).

Для поддержки многопроцессорной конфигурации контроллер прерываний в МП «Эльбрус» имеет распределенную структуру. Упрощенная схема контроллера прерываний для двухпроцессорной системы приведена на рисунке. Элементы системы прерываний взаимодействуют через аппаратно формируемые сообщения о прерываниях. Общая для процессора часть контроллера прерываний PREPIC (Processor Elbrus Interrut Controller) отвечает за доставку сообщения нужному CEPIC, при необходимости направляя сообщение в PREPIC другого процессора.

Внешнее прерывание, как правило, приходит в виде MSI-сообщения (Message Signaled Interrupt) вместе с потоком DMA-записей (операция прямого доступа в память – Direct Memory Access) от контроллера периферийных устройств, проходит процедуру трансляции и преобразуется в сообщение о прерывании.

Виртуализация системы прерываний

Виртуальная машина эмулирует аппаратные ресурсы, используя совокупность методов виртуализации аппаратуры ядра, памяти и компонентов ввода-вывода. Аппаратные ресурсы могут быть эмулированы программно или распределены между виртуальными машинами: системное ПО выделяет и подготавливает физические ресурсы для одной из виртуальных машин, в результате чего исполнение гостевого кода происходит на реальной аппаратуре (так называемое прямое исполнение).

Системное ПО, распределяющее физические ресурсы между виртуальными машинами, называют гипервизором (или монитором виртуальной машины). Традиционно архитектуру гипервизора относят к одному из двух типов: гипервизор первого типа исполняется независимым системным ПО, непосредственно работающим с аппаратными ресурсами (например, Xen [4]); гипервизор второго типа является расширением операционной системы, называемой хостовской ОС (например, KVM [5]).

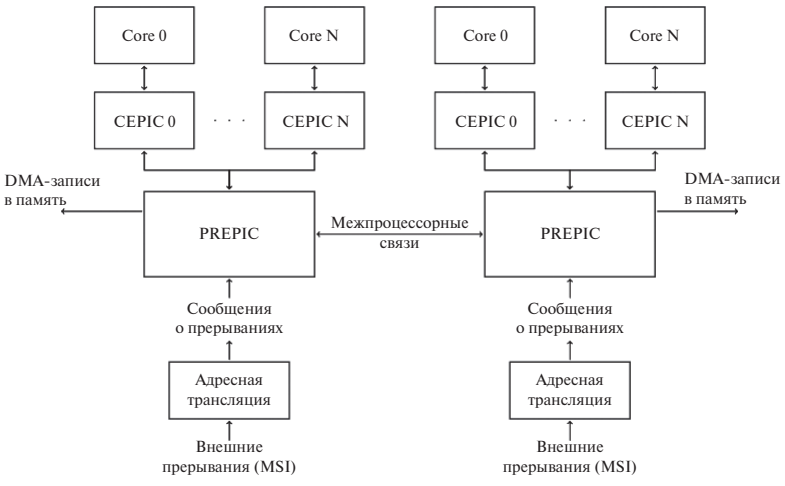

Рисунок. Упрощенная схема контроллера прерываний (EPIC) для двухпроцессорной системы: Core 0, Core N – ядра микропроцессора

Программная эмуляция всей машины крайне неэффективна. Успех имели подходы, при которых между виртуальными машинами распределялись ресурсы ядра и памяти, а остальное (в том числе система прерываний и компоненты ввода-вывода) эмулировалось программно [6].

Программная эмуляция ввода-вывода удобна тем, что все виртуальные машины могут иметь одинаковые компоненты (с точки зрения гостевой ОС) независимо от того, какая в действительности аппаратура используется. Однако современные вычислительные платформы включают архитектурную поддержку для виртуализации ввода-вывода и системы прерываний.

Взаимодействие с подсистемой ввода-вывода можно условно разделить на три категории:

1. Процессор может обращаться к регистрам внешнего устройства, отображенным на физическое адресное пространство MMIO.
2. Операции прямого доступа в память DMA позволяют внешним устройствам обращаться в память.
3. Прерывание позволяет внешним устройствам взаимодействовать с процессором.

Эти три группы тесно связаны функционально, поэтому при проектировании методов аппаратной поддержки виртуализации системы прерываний необходимо учитывать ограничения и специфику реализации компонентов ввода-вывода в МП семейства «Эльбрус».

Виртуализация системы прерываний программными средствами подразумевает следующее. Состояние гостевой системы прерываний (копии управляющих регистров) хранится в памяти машины. Обращения гостевого ядра к регистрам контроллера прерываний, а также ситуации, связанные с доставкой гостевых прерываний, перехватываются гипервизором и обрабатываются программно. Каждый такой перехват приводит к необходимости обращаться в память, что занимает длительное время. Доставка гостевых прерываний вместе с обработкой DMA-операций от внешних устройств являются достаточно затратным механизмом при программной виртуализации ввода-вывода [7, 8].

Одним из вариантов аппаратной поддержки виртуализации системы прерываний является введение дополнительных управляющих регистров контроллера прерываний, отражающих состояние гостевой системы прерываний (полностью или частично). Для доставки гостевых сообщений о прерываниях необходимо определять, на каком физическом ядре исполняется виртуальное ядро, которому предназначено прерывание. Эту функцию выполняет аппаратная таблица соответствия гостевых и физических ядер. Состояние таблицы соответствия должно быть одинаковым во всех процессорах многопроцессорной системы, поэтому вводятся аппаратные механизмы управления таблицей, обеспечивающие корректность работы в таких системах.

Предлагаемые принципы реализации аппаратной поддержки виртуализации системы прерываний являются частью общего подхода к созданию средств виртуализации для архитектуры «Эльбрус» [9].

Аппаратная поддержка виртуализации в контроллере прерываний

Гостевой набор управляющих регистров контроллера прерываний

Перехват обращения к регистрам контроллера прерываний можно не делать, если реализовать в аппаратуре дополнительный (гостевой) набор управляющих регистров (по аналогии с теневыми регистрами в аппаратуре ядра), который будет использоваться гостевым ядром, активным на данном физическом ядре в текущий момент. При снятии гостевого ядра гипервизор сохраняет состояние гостевых управляющих регистров в памяти, при его постановке – восстанавливает значения регистров в соответствии с сохраненным состоянием.

Введение гостевого набора регистров осложняет процедуры переключения гостей из-за необходимости сохранять и восстанавливать состояния регистров, однако позволяет обрабатывать гостевые обращения к регистрам контроллера прерываний без перехвата и обращения в память. Когда в аппаратуре появляются гостевые сообщения о прерываниях, они сопровождаются номером гостевой машины для изоляции гостевых прерываний от прерываний хоста и друг от друга.

Таблица соответствия физических ядер и активных гостевых ядер

Для определения, на каком физическом ядре при доставке располагается гостевое сообщение о прерывании, в аппаратуре реализована таблица соответствия DAT (destination address table), хранящая для каждого физического ядра номер активного в данный момент гостевого ядра. Таблица соответствия расположена в PREPIC и управляется гипервизором через обращение к соответствующему регистру CEPIC: запись в регистр формирует управляющее сообщение об изменении состояния гостевого ядра. Аппаратура позволяет изменять состояние гостевого ядра в DAT для конкретного физического ядра, не затрагивая работу остальных ядер. Для корректной работы в рамках многопроцессорной системы состояние таблицы соответствия должно быть одинаковым во всех процессорах

многопроцессорной системы, это свойство реализуется через аппаратно формируемые управляющие сообщения.

Если гостевое ядро, которому предназначено сообщение о прерывании, активно, то номера гостевой машины и гостевого ядра преобразуются в физический номер ядра. Если гостевое ядро не активно (отложено), необходимо зафиксировать гостевое прерывание в памяти.

Доставка прерываний отложенному гостевому ядру

Если гостевое ядро не активно (отложено), сообщение о прерывании помещается в буфер гостевых прерываний BGI (buffer for guest interrupts), который расположен в PREPIC и освобождается аппаратно формируемыми атомарными DMA-операциями записи в память. Записи в память, связанные с доставкой прерываний отложенному гостю, должны формироваться аппаратно, поскольку в противном случае возможен дедлок, то есть:

- поток прерываний, предназначенных отложенным гостевым ядрам, может значительно превосходить пропускную способность ядра по обращению к управляющим регистрам контроллера прерываний (контроллер прерываний одновременно обрабатывает не более одного запроса к регистрам за раз);
- из-за вышеуказанной проблемы возможна ситуация переполнения очереди прерываний для гостевых ядер, ожидающих записи в память; как следствие, PREPIC не сможет принять новое сообщение о внешнем прерывании;
- вследствие вышесказанного и упорядоченности MSI-сообщений, ответов на запросы к регистрам внешних устройств и DMA-обращений от внешних устройств ядро не получит ответ на возможное чтение регистра внешних устройств; последующие запросы в пространство ввода-вывода и к регистрам контроллера прерываний заблокируются;
- блокировка запросов в этом случае будет продолжаться, пока прерывания гостевому ядру не будут зафиксированы в памяти.

Изменение состояния гостевого ядра в DAT

В описываемой распределенной системе неизбежны ситуации гонок между сообщениями о прерываниях, управляющими сообщениями и DMA-операциями, связанными с доставкой прерываний отложенным гостевым ядрам.

После снятия гостевого ядра с физического (при изменении строки в DAT) в многопроцессорной системе остаются сообщения о прерываниях, для которых преобразование гостевого номера в физический уже произошло. Такое прерывание может быть потеряно, если значение регистра (при сохранении его состояния в память) будет прочитано до того, как прерывание дойдет до CEPIC.

Похожая ситуация возникает при постановке гостевого ядра – после изменения строки в DAT в системе остаются предназначенные этому гостевому ядру прерывания, DMA-записи для доставки которых еще не завершились. При восстановлении значения регистра из памяти запрос на чтение от гипервизора может обогнать указанную DMA-запись, а прерывание останется в памяти до следующего переключения данного гостевого ядра.

Для исключения таких ситуаций аппаратура предоставляет системному ПО возможность точно определить моменты:

- доставки в целевой CEPIC всех сообщений о прерываниях, прошедших преобразование гостевого номера ядра в физический до того, как в DAT изменилось состояние строки;
- завершения всех связанных с доставкой прерывания отложенному гостевому ядру DMA-операций, сформированных до того, как в DAT изменилось состояние строки.

Заключение

В статье описана архитектура разработанного распределенного контроллера прерываний для МП семейства «Эльбрус». Изложены принципы реализации аппаратной поддержки виртуализации системы прерываний, благодаря которой исключаются ситуации перехвата при обработке обращений гостевого ядра к управляющим регистрам контроллера прерываний и доставке гостевых прерываний за счет введения дублирующего набора регистров и следующих аппаратных механизмов:

- определения физического ядра, на котором активно целевое гостевое ядро;
- доставки прерываний отложенному гостевому ядру через атомарные DMA-обращения;
- обеспечения корректности при переключении гостевых ядер.

Следует отметить, что в результате введения указанных механизмов увеличиваются площадь кристалла, занимаемого контроллером прерываний, и время, затрачиваемое процедурами переключения гостей (за счет сохранения/восстановления состояния управляющих регистров), но, несмотря на это, сохраняется целесообразность описанных усовершенствований.

В настоящий момент разработанный контроллер прерываний с реализованной аппаратной поддержкой виртуализации проходит стадию автономной верификации.

СПИСОК ЛИТЕРАТУРЫ

1. Intel Virtualization Technology for Directed I/O, Architecture Specification. Intel, 2016.
2. ARM Generic Interrupt Controller Architecture Specification v2.0. ARM, 2013.
3. Introduction of AMD Advanced Virtual Interrupt Controller. XenSummit, 2012.
4. Pratt Ia., Fraser K., Hand S., Limpach Ch., Warfield A., Magenheimer D., Nakajima J., Mallick A. Xen 3.0 and the art of virtualization. In Proc. of the 2005 Ottawa Linux Symposium (OLS), 2005.
5. Kivity A. KVM: The linux virtual machine monitor. In Proc. of the 2007 Ottawa Linux Symposium (OLS), July 2007, pp. 225–230.
6. Bugnion E., Devine S., Rosenblum M., Sugerman J., Wang E. Y. Bringing virtualization to the x86 architecture with the original VMware workstation. ACM Transactions on Computer Systems, 2012, vol. 30, no. 4, pp. 12:1–12:51.
7. Adams K., Agesen O. A comparison of software and hardware techniques for x86 virtualization. ACM ASPLOS'06, San Jose, California, USA, oct. 21–25, 2006.
8. Gordon A., Amit N., Har'El N., Ben-Yehuda M., Landau A., Schuster A., Tsafrir D. Eli: bare-metal performance for i/o virtualization. ACM SIGARCH Computer Architecture News, 2012, vol. 40 (1), pp. 411–422.
9. Знаменский Д. В. Выбор вариантов реализации средств аппаратной поддержки виртуализации архитектуры «Эльбрус» // Вопросы радиоэлектроники. 2014. Вып. 3. С. 64–73.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Деменко Роман Витальевич, аспирант МФТИ (ГУ), инженер, АО «МЦСТ», 119334, Москва, ул. Вавилова, д. 24, тел.: 8 (963) 752-00-16, e-mail: roman.dmnk@gmail.com.
Трофимов Валентин Борисович, ведущий инженер-конструктор, АО «МЦСТ», 119334, Москва, ул. Вавилова, д. 24, тел.: 8 (903) 975-10-98, e-mail: trovb@mcst.ru.

For citation: Demenko R. V., Trofimov V. B. Hardware virtualization support for interrupt controller in Elbrus series processors. Voprosy radioelektroniki, 2018, no. 2, pp. 40–44.

R. V. Demenko, V. B. Trofimov

HARDWARE VIRTUALIZATION SUPPORT FOR INTERRUPT CONTROLLER IN ELBRUS SERIES PROCESSORS

Modern multi-core processors implement hardware virtualization support in order to reduce the corresponding overhead. In contrast to the CPU core, successful implementations of the hardware support for I/O virtualization were introduced recently. One of the mechanisms that requires a hardware-assisted virtualization is a guest interrupt delivery to its target virtual core without involving the hypervisor. The paper describes the architecture of the distributed interrupt controller for Elbrus series processors and basic principles for hardware virtualization support implementation for interrupt controller. We propose to implement a guest set of interrupt controller control registers and provide a virtual interrupt delivery technique using the hardware table of correspondence between virtual and physical cores. We describe mechanisms to provide the correct operation of the considered interrupt system with implemented hardware virtualization support within the four-processor system.

Keywords: Elbrus, virtualization, hypervisor, interrupt controller, virtual interrupts.

REFERENCES

1. Intel Virtualization Technology for Directed I/O, Architecture Specification. Intel, 2016.
2. ARM Generic Interrupt Controller Architecture Specification v2.0. ARM, 2013.
3. Introduction of AMD Advanced Virtual Interrupt Controller. *XenSummit,* 2012.
4. Pratt Ia., Fraser K., Hand S., Limpach Ch., Warfield A., Magenheimer D., Nakajima J., Mallick A. Xen 3.0 and the art of virtualization. *In Proc. of the 2005 Ottawa Linux Symposium (OLS),* 2005.
5. Kivity A. KVM: The linux virtual machine monitor. *In Proc. of the 2007 Ottawa Linux Symposium (OLS),* July 2007, pp. 225–230.
6. Bugnion E., Devine S., Rosenblum M., Sugerman J., Wang E. Y. Bringing virtualization to the x86 architecture with the original VMware workstation. *ACM Transactions on Computer Systems,* 2012, vol. 30, no. 4, pp. 12:1–12:51.
7. Adams K., Agesen O. A comparison of software and hardware techniques for x86 virtualization. *ACM ASPLOS'06,* San Jose, California, USA, oct. 21–25, 2006.
8. Gordon A., Amit N., Har'El N., Ben-Yehuda M., Landau A., Schuster A., Tsafrir D. Eli: bare-metal performance for i/o virtualization. *ACM SIGARCH Computer Architecture News,* 2012, vol. 40 (1), pp. 411–422.
9. Znamenskiy D. V. Alternatives of hardware virtualization support inplementation for Elbrus processor architecture. *Voprosy radioelektroniki,* 2014, no. 3, pp. 64–73 (In Russian).

AUTHORS

Demenko Roman, graduate student, MIPT, engineer, JSC MCST, 24, ulitsa Vavilova, Moscow, 119334, Russian Federation, tel.: +7 (963) 752-00-16, e-mail: roman.dmnk@gmail.com.
Trofimov Valentin, leading design engineer, JSC MCST, 24, ulitsa Vavilova, Moscow, 119334, Russian Federation, tel.: +7 (903) 975-10-98, e-mail: trovb@mcst.ru.