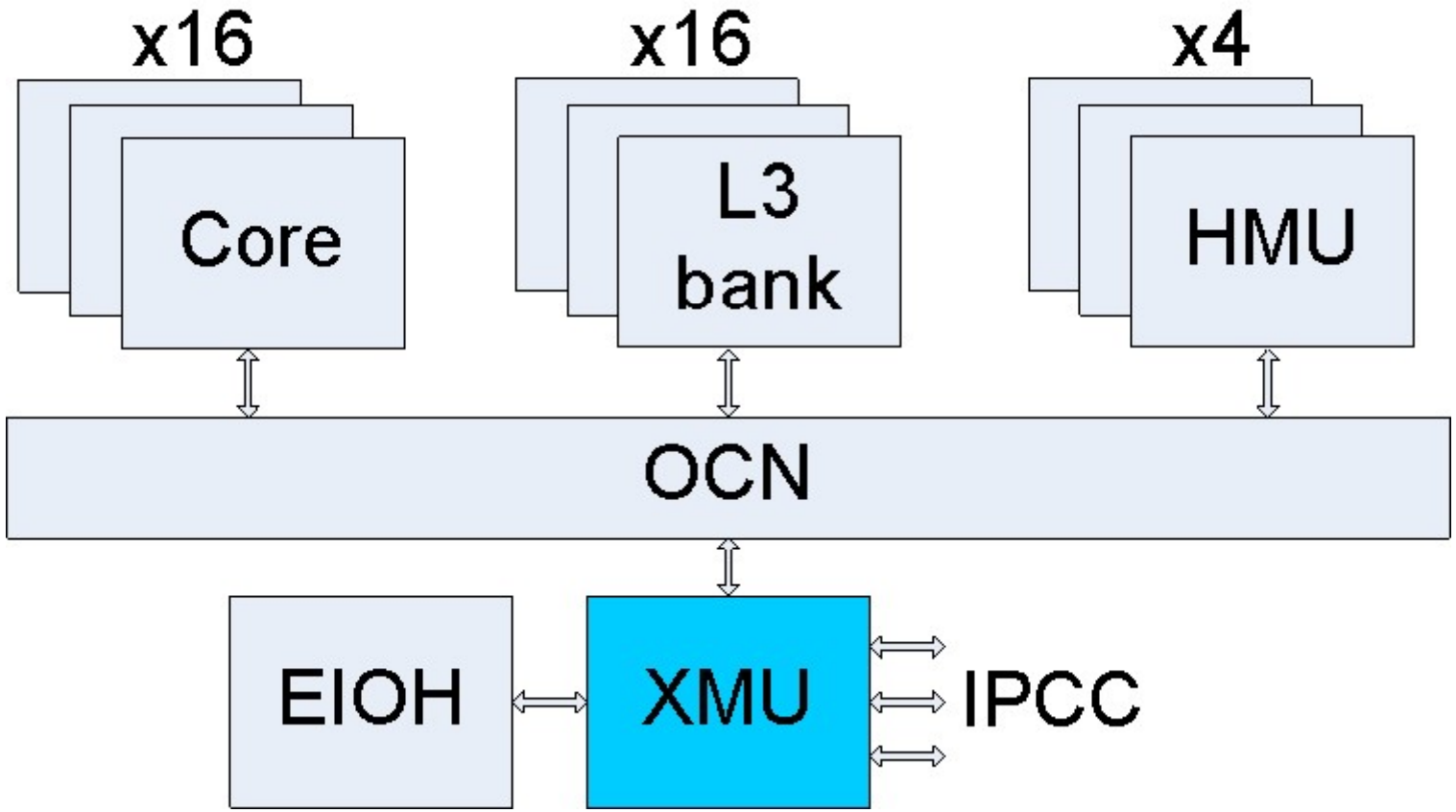
Виртуализация системы прерываний в МП архитектуры Эльбрус

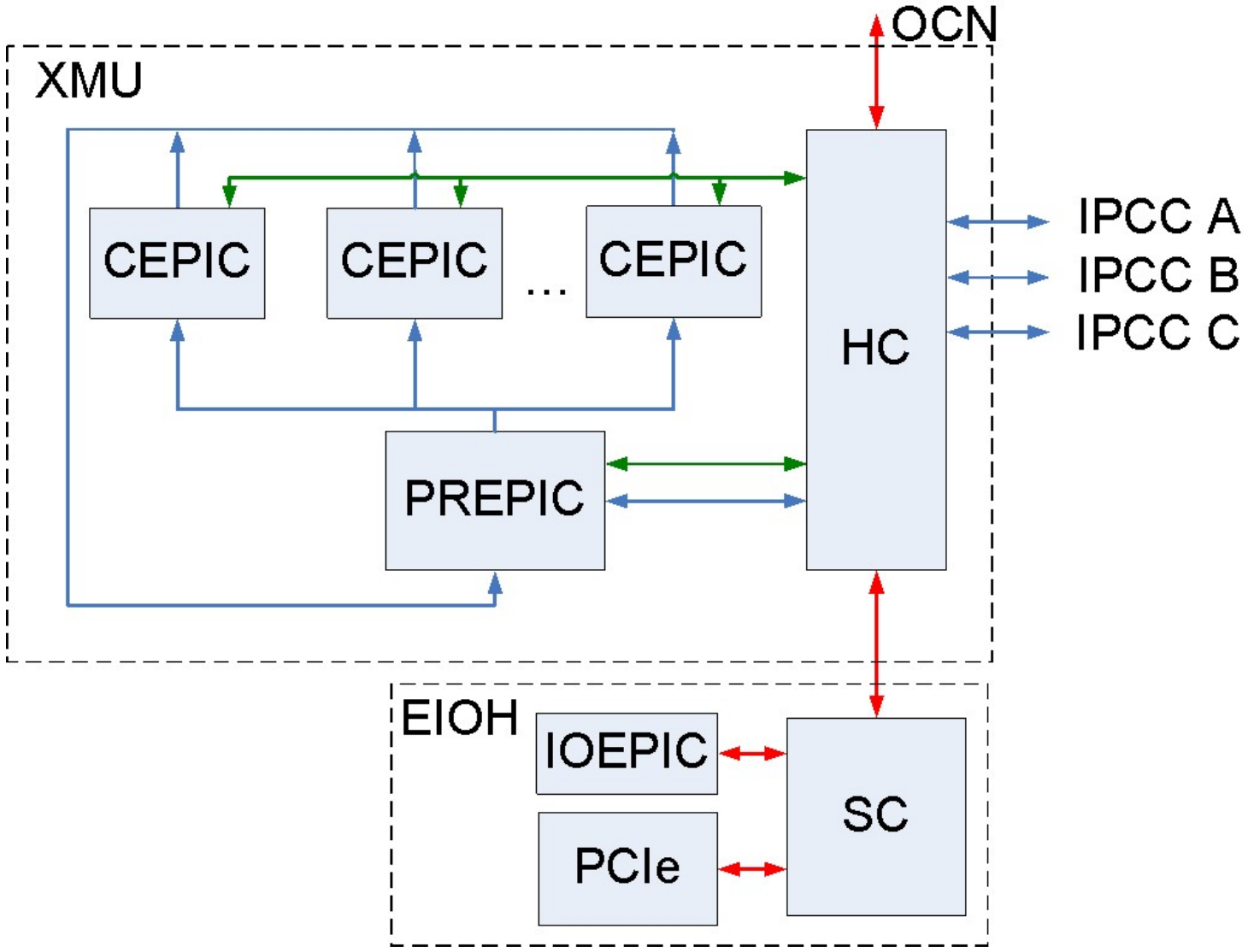
План доклада

- Структура системы прерываний
- Виртуальные прерывания в проектах Эльбрус-16С, Эльбрус-2С3
- Виртуальные прерывания в Intel VT-d

XMU в составе E16C

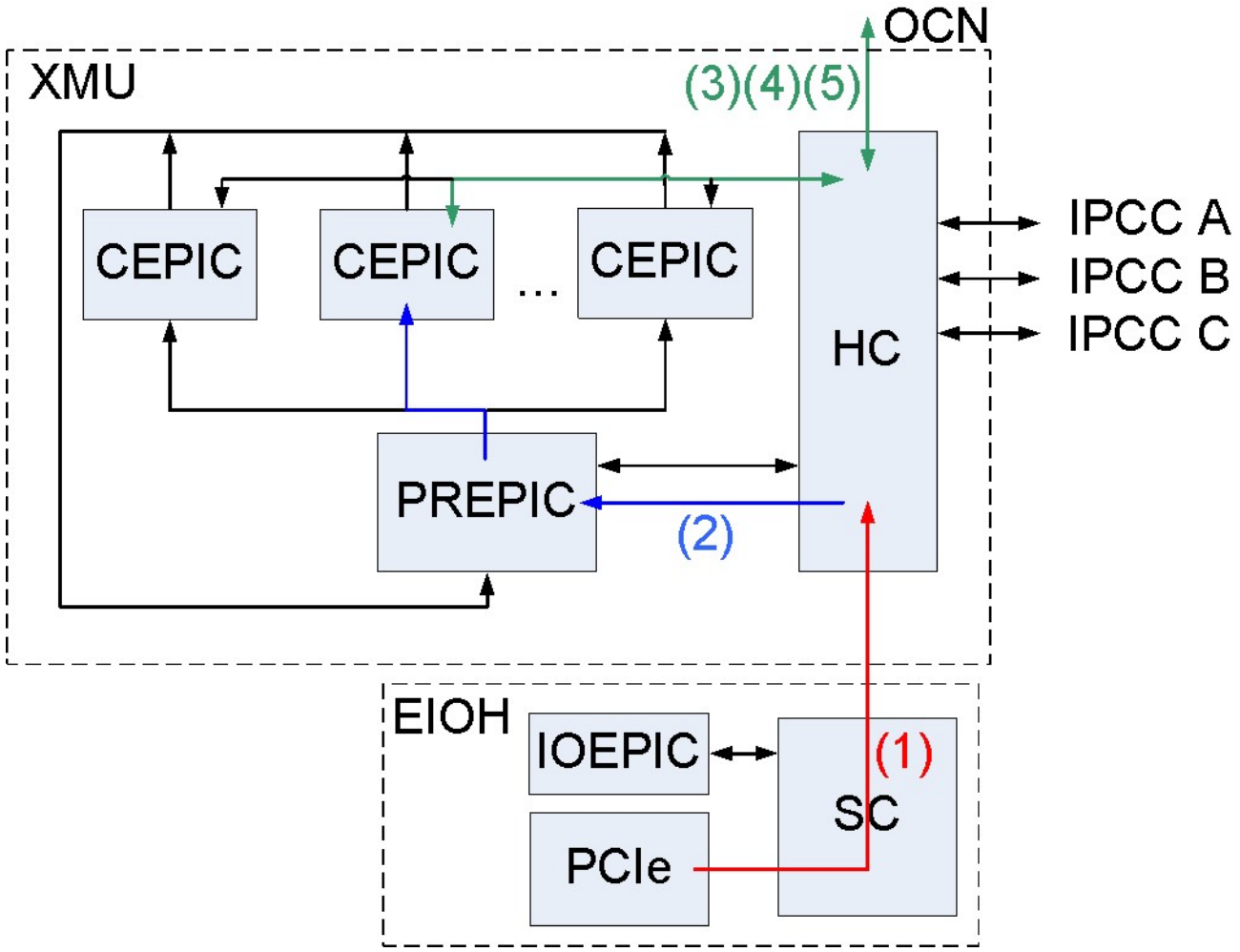


Компоненты EPIC



Источники прерываний

- внешние устройства (MSI/MSI-x или через IOEPIC)
- запись в регистр CEPIC (межпроцессорные прерывания)
- локальные таймеры в CEPIC
- системные (аварийное, перегрев, etc.)



(1) Message Signaled Interrupt (MSI) (DMA)
(2) Сообщение о прерывании (сообщение формата EPIC)
(3) Сигнал в аппаратуру ядра
о наличии необработанного прерывания (сигнал)
(4) Определение вектора прерывания (обращение к регистрам)
(5) End of interrupt (EOI) (обращение к регистрам)

Регистры доставки прерываний

- **CIR - current interrupt request** – наиболее приоритетное из необработанных прерываний
- **PMIRR – pending maskable interrupts** – отложенные прерывания (1 бит на прерывание)
- **CPR – core priority** – текущий приоритет задачи
- **VECT_INTA** – начало обработки прерывания, определение вектора, освобождение CIR
- **EOI – end of interrupt** – завершение обработки прерывания
- **ICR – interrupt command** – запись формирует межпроцессорное прерывание

Резюме: система прерываний

- набор регистров контроллера прерываний
- сигналы в процессорные ядра о наличии необработанных прерываний
- запросы на чтение/запись регистров от процессорных ядер
- поток DMA записей от внешних устройств, часть из которых - внешние прерывания
- сообщения специального формата, программно не видны
- локальные таймеры

Виртуализация

Виртуализация через программную эмуляцию

- 1999 - VMWare Workstation 1.0
- 2003 - Xen 1.0

Аппаратная поддержка виртуализации (core, memory)

- Intel VT-x 2005
- AMD-V 2006

Аппаратная поддержка виртуализации (I/O, interrupts)

- Intel VT-d 2012
- AMD-Vi 2012

Виртуализация системы прерываний

- Состояние регистров виртуального контроллера прерываний
- Запросы к регистрам
- Доставка внешних прерывания

Без аппаратной поддержки системы прерываний:

- программная модель контроллера прерываний
- запросы к регистрам и доставка прерываний – через перехват и эмуляцию (trap-and-emulate)

Цель введения аппаратной поддержки:

избавиться от необходимости перехвата (#vmexit).

Elbrus Programmable Interrupt Controller

Виртуализация EPIC: гостевой набор регистров

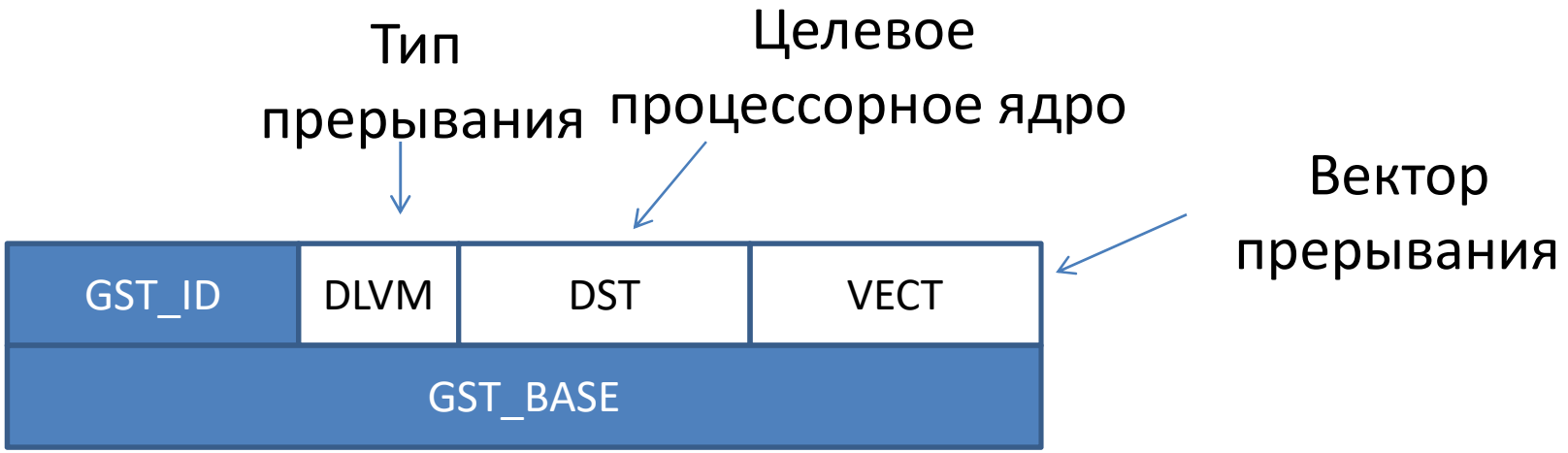
- Дополнительный набор регистров CEPIC – (для виртуального ядра)
- Копия регистров CEPIC – в оперативной памяти
- Сохраняем/восстанавливаем состояние виртуального CEPIC в/из памяти при постановке/снятии/миграции виртуального ядра.

Виртуальное ядро	**Актуальное состояние виртуального CEPIC**
активно	на регистрах
отложено	в памяти

Виртуализация EPIC: сообщения о прерываниях

В аппаратуре сообщения о прерываниях сопровождаются

- **GST_ID** - идентификатором виртуальной машины
- **GST_BASE** - базовым адресом для записи прерываний в память

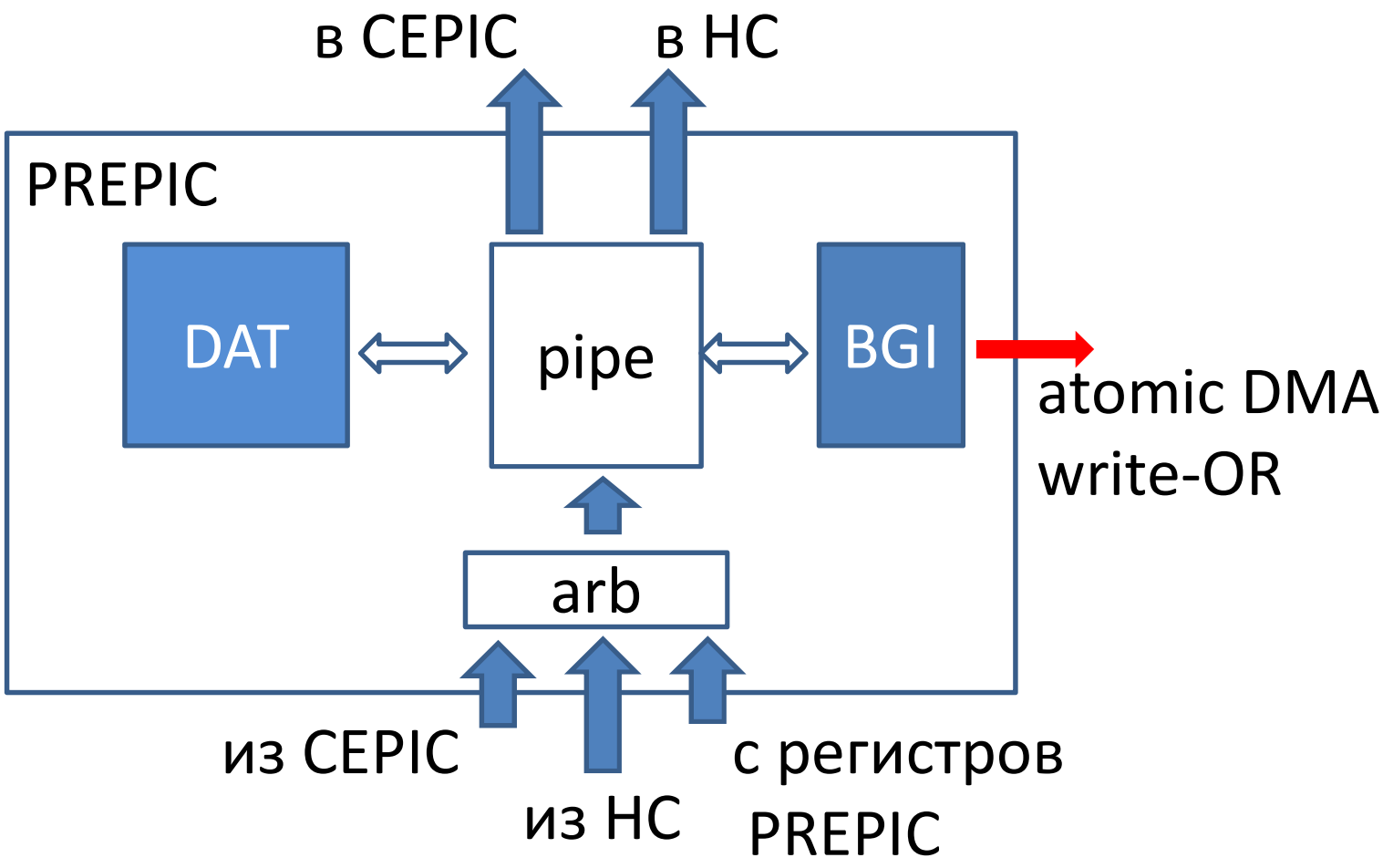


Виртуализация EPIC: Destination Address Table

PREPIC хранит таблицу соответствия виртуальных и физических ядер

Index (phys. CEPIC_ID)	val	guest ID	guest DST (virtual CEPIC_ID)
0	1	0x34	4
1	0	-	-
2	1	0x015	1
…	…		
63	1	0x34	9

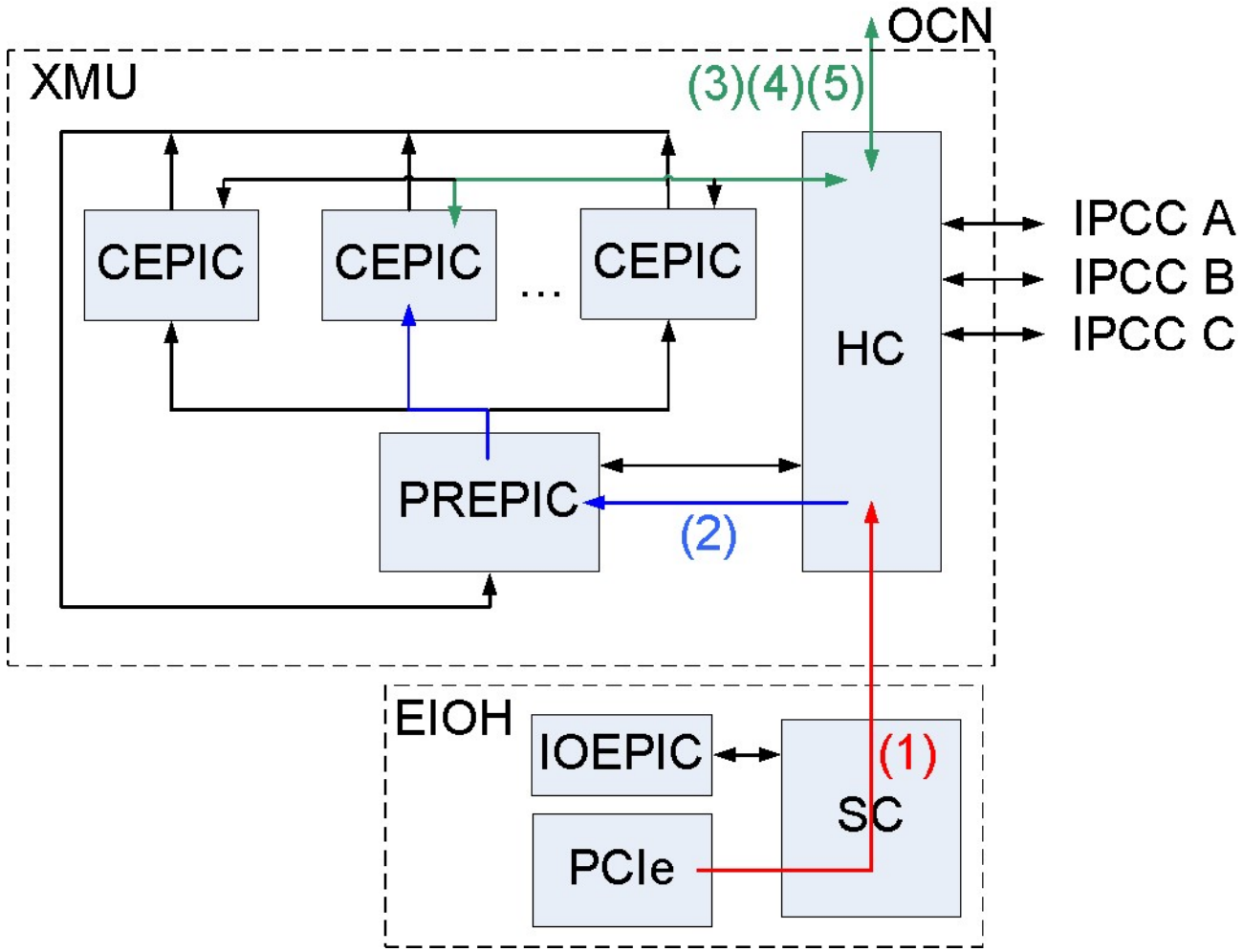
Ассоциативный поиск по {guest ID, guest DST}



HIT -> виртуальное ядро активно, подменяем DST

MISS -> виртуальное ядро отложено, формируем DMA

Где **guest ID** и **guest_base** ?



Где **guest ID** и **guest_base ?**

Для **межпроцессорных прерываний** - с регистров CEPIC (настраиваются гипервизором).

Для **внешних прерываний –** как результат трансляции в IOMMU (настраивается гипервизором)

(bus,dev,func) -> {enbl, guest_ID, guest_base}

Сложности подхода

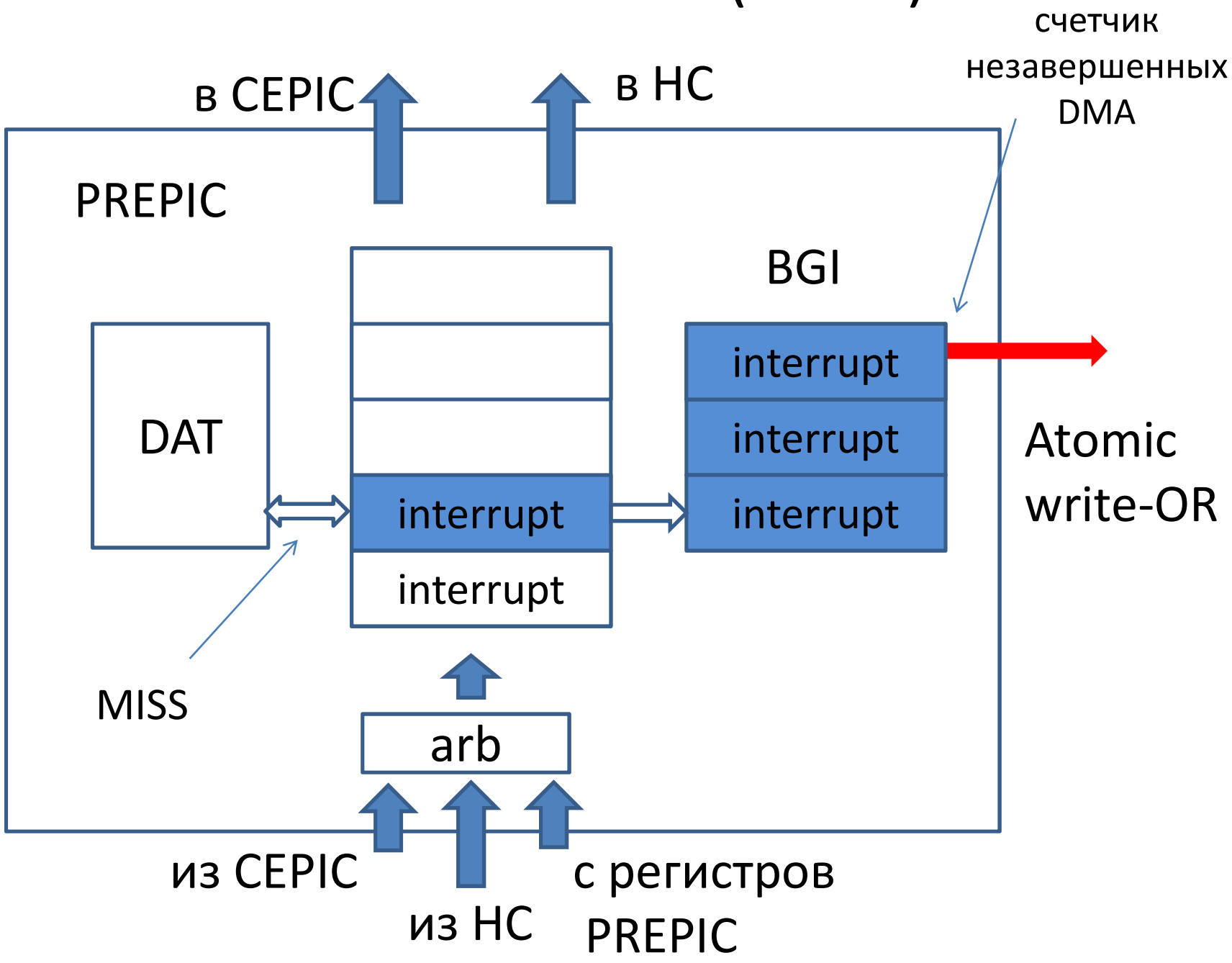
- Сохранение/восстановление состояния CEPIC – не атомарный процесс
- Состояние DAT должно быть одинаковым в всех PREPIC’ах многопроцессорной системы, изменение строки в DAT – не атомарный процесс
- Виртуальное прерывание может прийти в любой момент

Сложности подхода

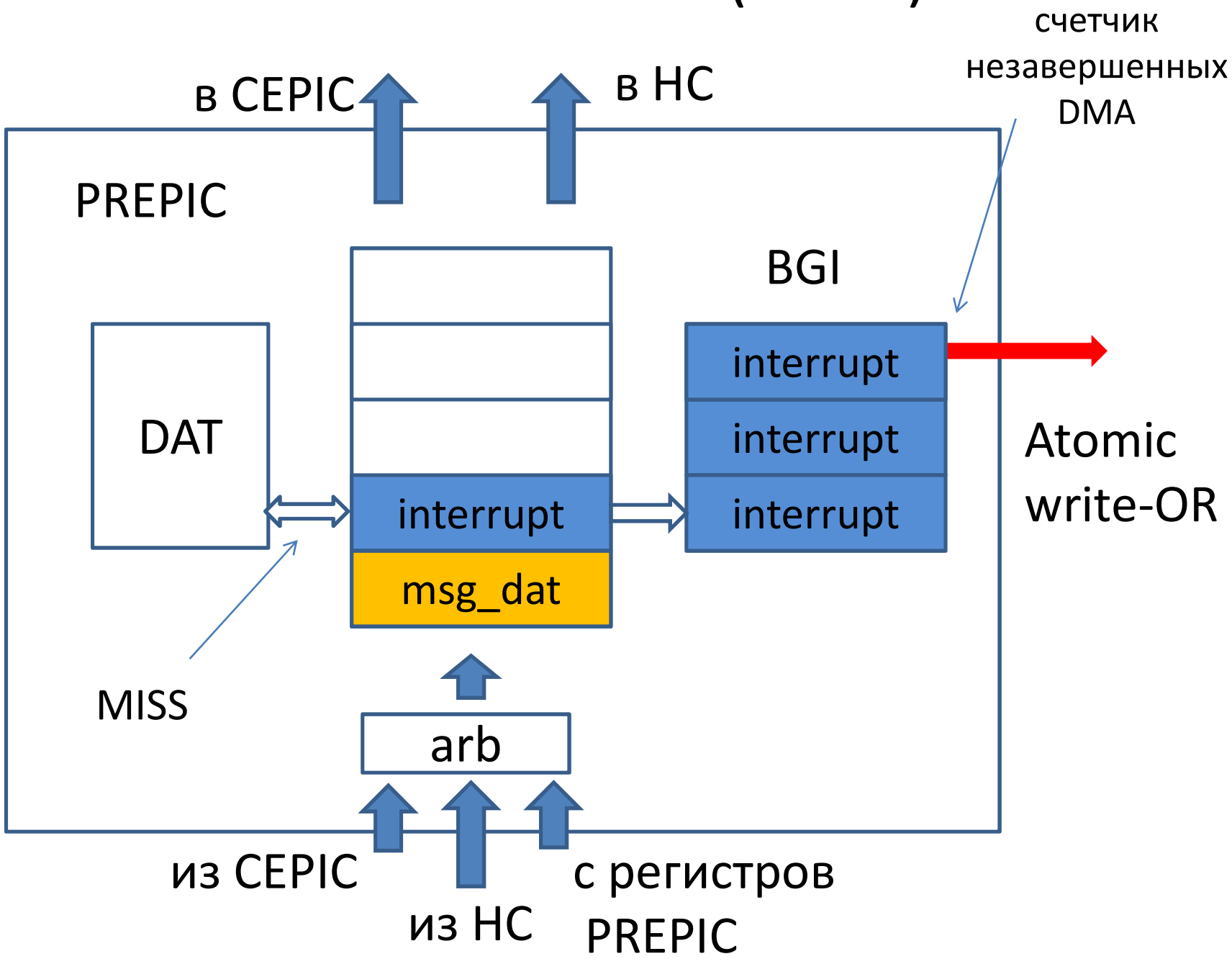
- В процессе снятия гостя после вычеркивания строки в DAT в системе остаются сообщения о прерываниях, которые уже получили HIT в PREPIC. Эти прерывания могут быть потеряны, если гипервизор начнет откачивать состояние в память до того, как прерывание дойдет до CEPIC.

- В процессе постановки гостя после обновления строки в DAT в системе остаются прерывания в виде незавершенных записей в память. Эти прерывания могут быть потеряны если гипервизор начнет восстанавливать состояние из памяти до того, как прерывания будут записаны в память.

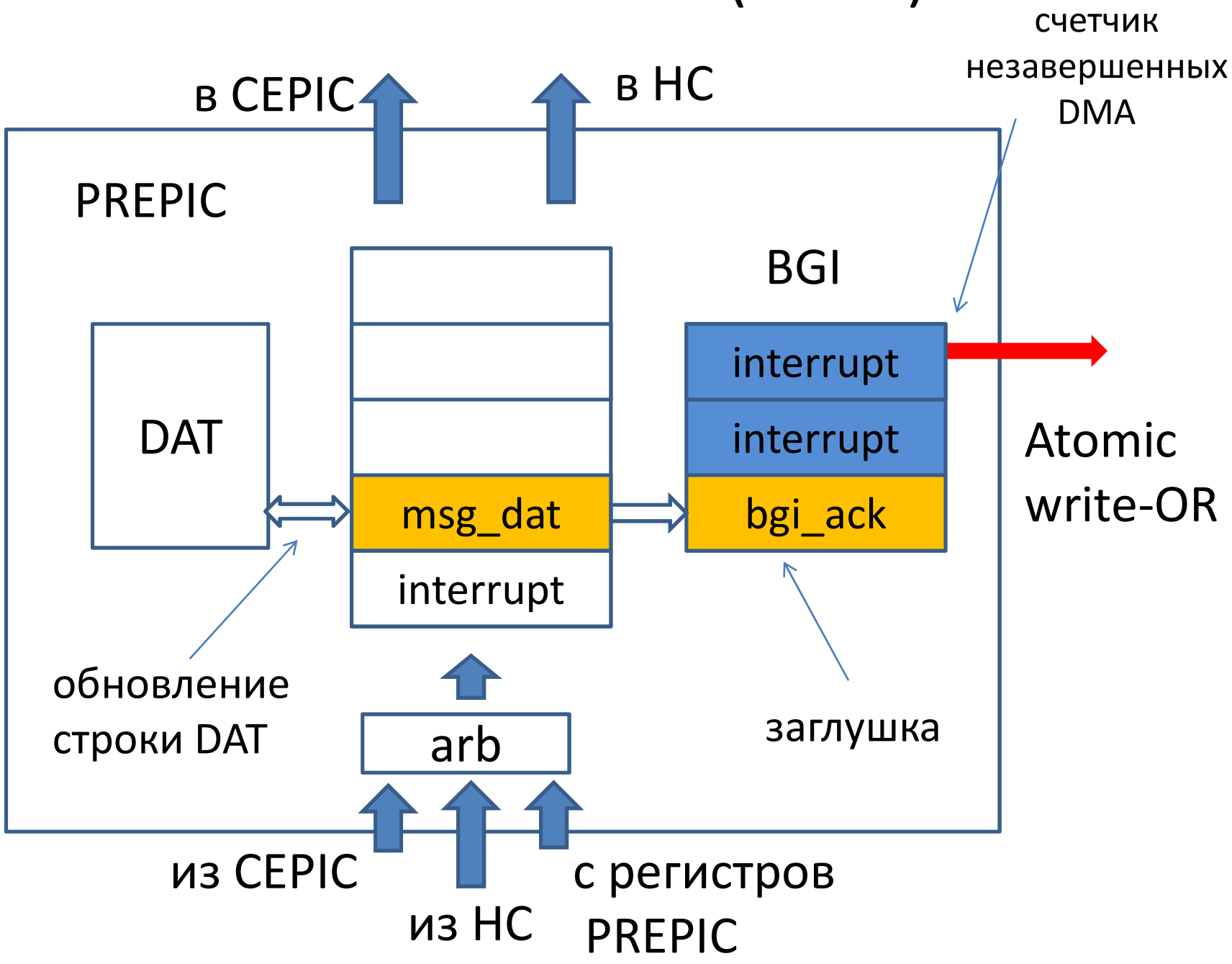
Постановка гостя (в DAT)



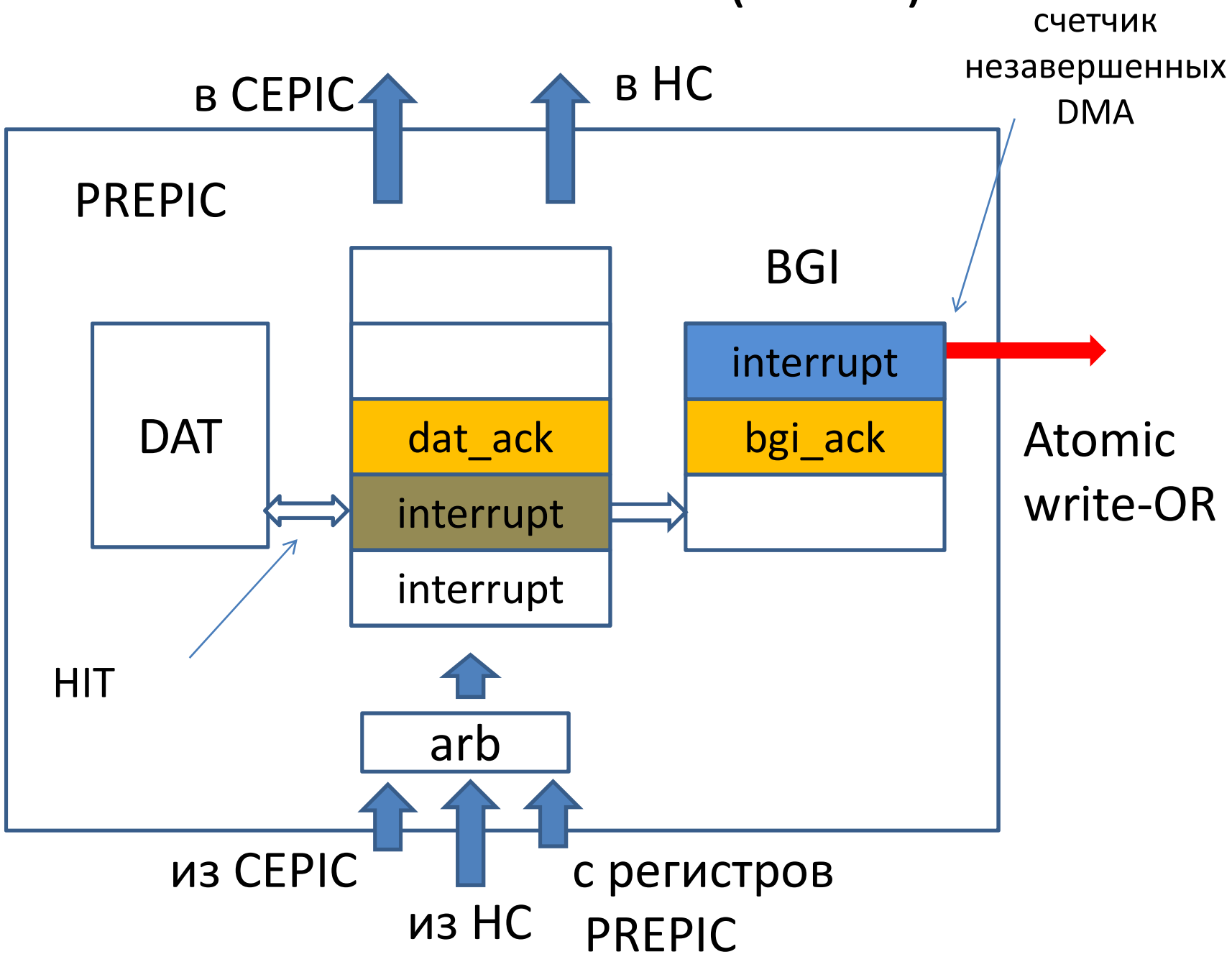
Постановка гостя (в DAT)



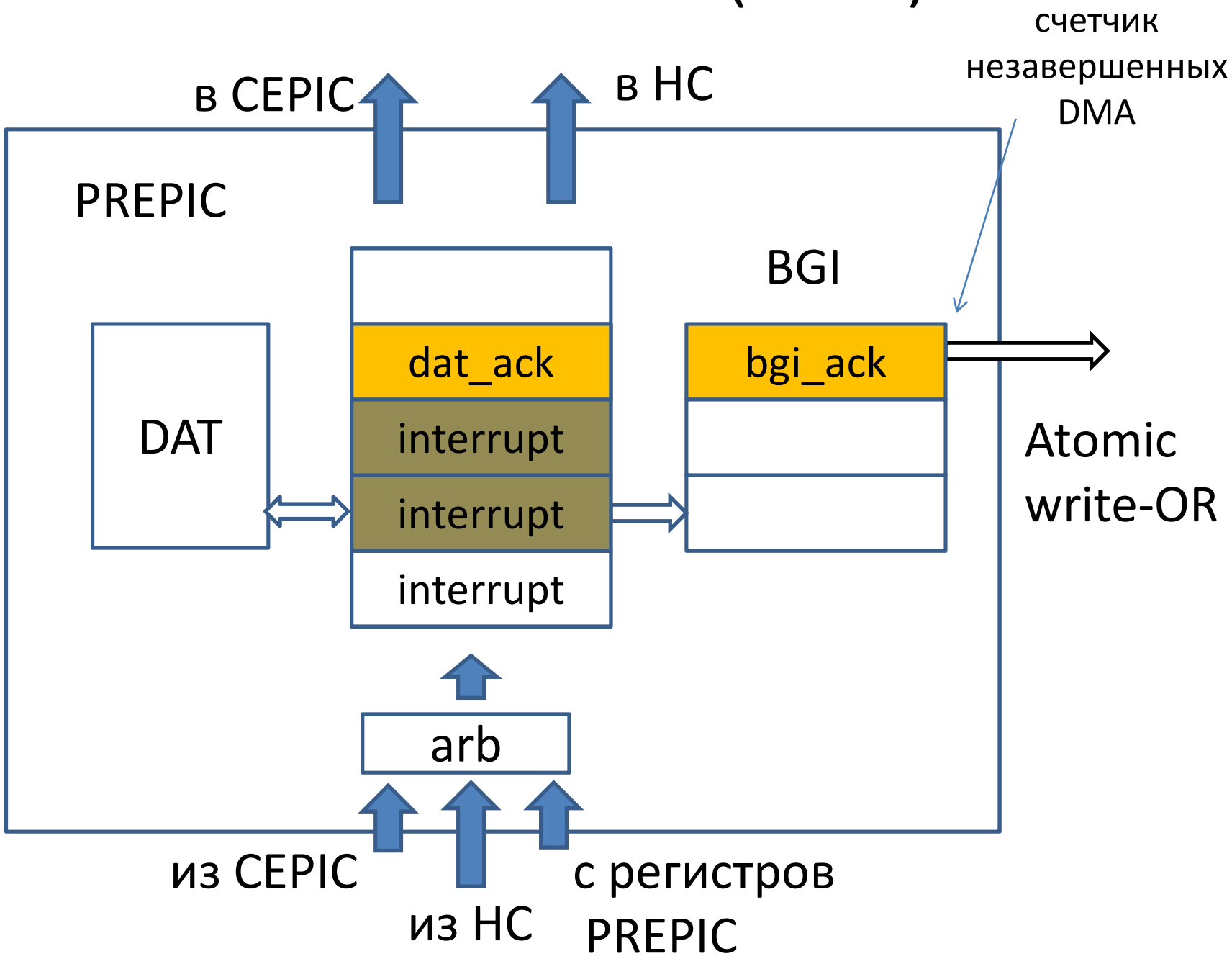
Постановка гостя (в DAT)



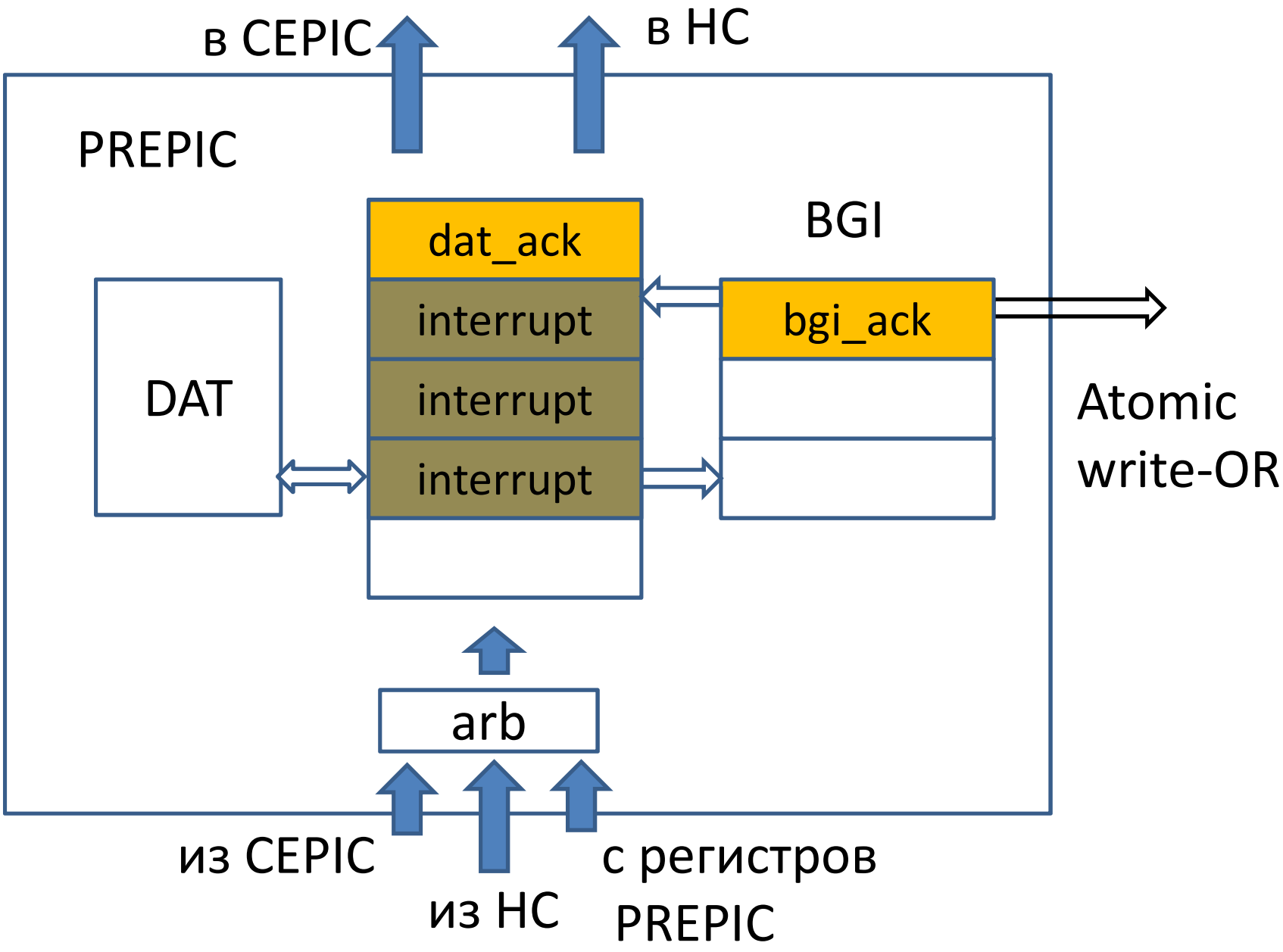
Постановка гостя (в DAT)



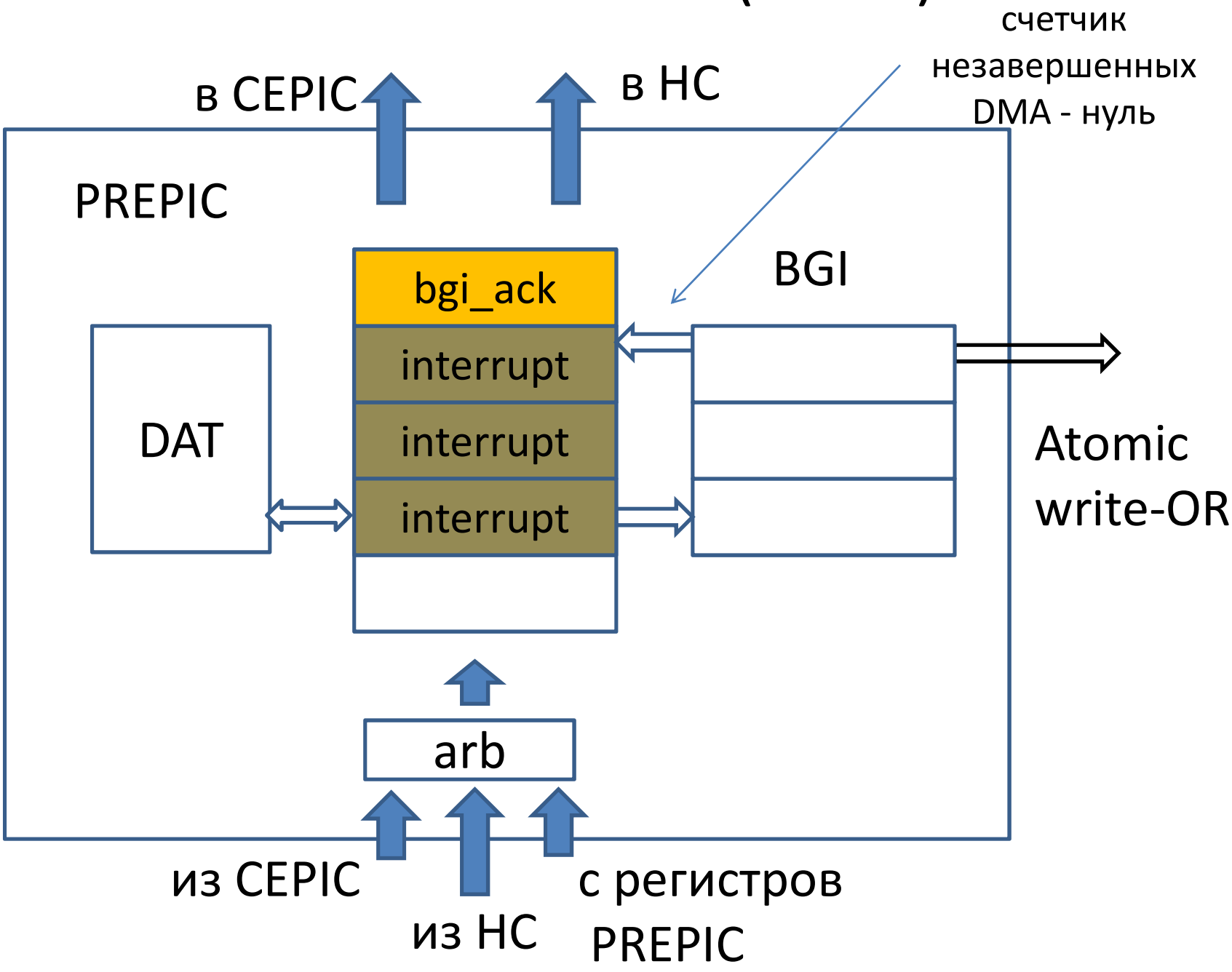
Постановка гостя (в DAT)



Постановка гостя (в DAT)



Постановка гостя (в DAT)



Изменение строки в DAT

Гипервизор

- запись в регистр CEPIC
- чтение в цикла, ожидание сброса статуса

CEPIC

- формируется сообщение msg_dat, выставляет бит статуса
- собирает по 2 квитанции от каждого PREPIC, сбрасывает бит статуса

Сброс статуса гарантирует завершение переходных процессов в EPIC.

Доступ к регистрам

CEPIC

- Гостевой набор регистров CEPIC доступен по тем же физ.адресам, что и основной набор. Запрос содержит признак хост/гость.
- Страница, на которую отображены регистры CEPIC, напрямую маппируется для гостя.
- Гиперпривилегированные регистры выделены на отдельную страницу, не маппируются для гостя

PREPIC

- Регистры PREPIC располагаются на отдельной странице и не маппируются для гостя. Гостевые обращения в PREPIC перехватываются, функциональность эмулируется программно.

IOEPIC

- В IOEPIC для каждого входа прерывания предусмотрен набор регистров. Каждый набор вынесен в отдельную страницу, и может быть отмаппирован независимо от других.

Резюме: виртуальные прерывания в EPIC

- Дополнительный набор регистров CEPIC
- Копия регистров CEPIC в оперативной памяти
- Программное сохранение/восстанавление состояния CEPIC в/из памяти при постановке/снятии виртуального ядра.
- Аппаратура для поиска физического ядра, соответствующего целевому виртуальному ядру. Перенаправление прерывания либо в CEPIC на регистры, либо в память.
- Аппаратура гарантирует отсутствие гонок между прерываниями/обращениями к регистрам и в память.
- Трансляция сообщений о внешних прерываниях в IOMMU.

Виртуальные прерывания (на примере Intel VT-d)

EPIC vs. APIC

- схожее разбиение на функциональные компоненты: CEPIC – LAPIC, IOEPIC – IOAPIC
- отличается семантика регистров
  - схема приоритетов и выбор наиболее приоритетного прерывания
  - работа с вложенными прерываниями
  - работа с уровневыми прерываниями
- отказались от устаревшей функциональности
- часть функциональности отдали ПО

Виртуальные прерывания (на примере Intel VT-d)

- **vAPIC** – виртуальный контроллер прерываний
- **vAPIC backing page** – копия регистров в оперативной памяти, маппируется как гостевой APIC_BASE
- гостевые обращения в APIC сводятся к обращениям в память, не вызывают перехват

APIC-write emulation

Гостевые обращения в APIC сводятся к обращениям в память, не вызывают перехват.

Аппаратура эмулирует побочные эффекты при обращении к отдельным регистрам (vIRR, vISR, vICR, vEOI).

Interrupt posting

Posted Interrupt Descriptor (PID) – область оперативной памяти для временного хранения гостевых прерываний.

Поля:

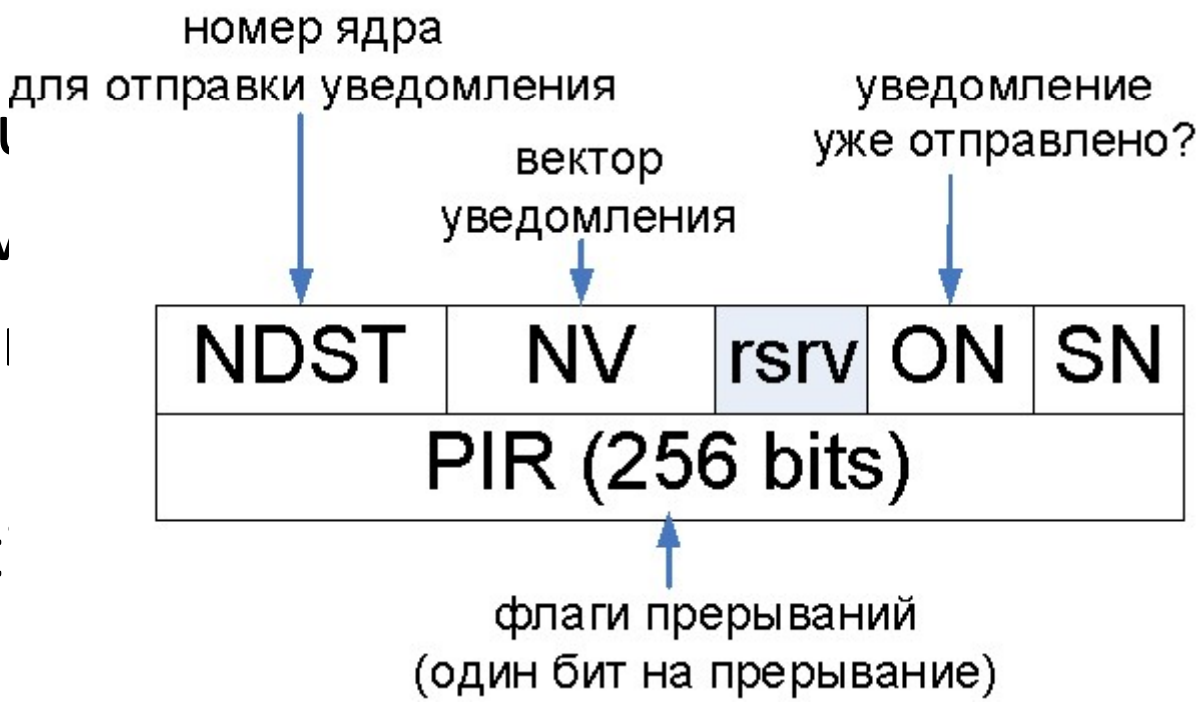
PIR - Posted Interr

NV – Notification V

NDST – Notificatio

ON – Outstanding

SN – Suppress Not



Инжектирование прерываний: Posted Interrupts

Гипервизор (Core0) инжектирует гостевое прерывание гостю (Core1).

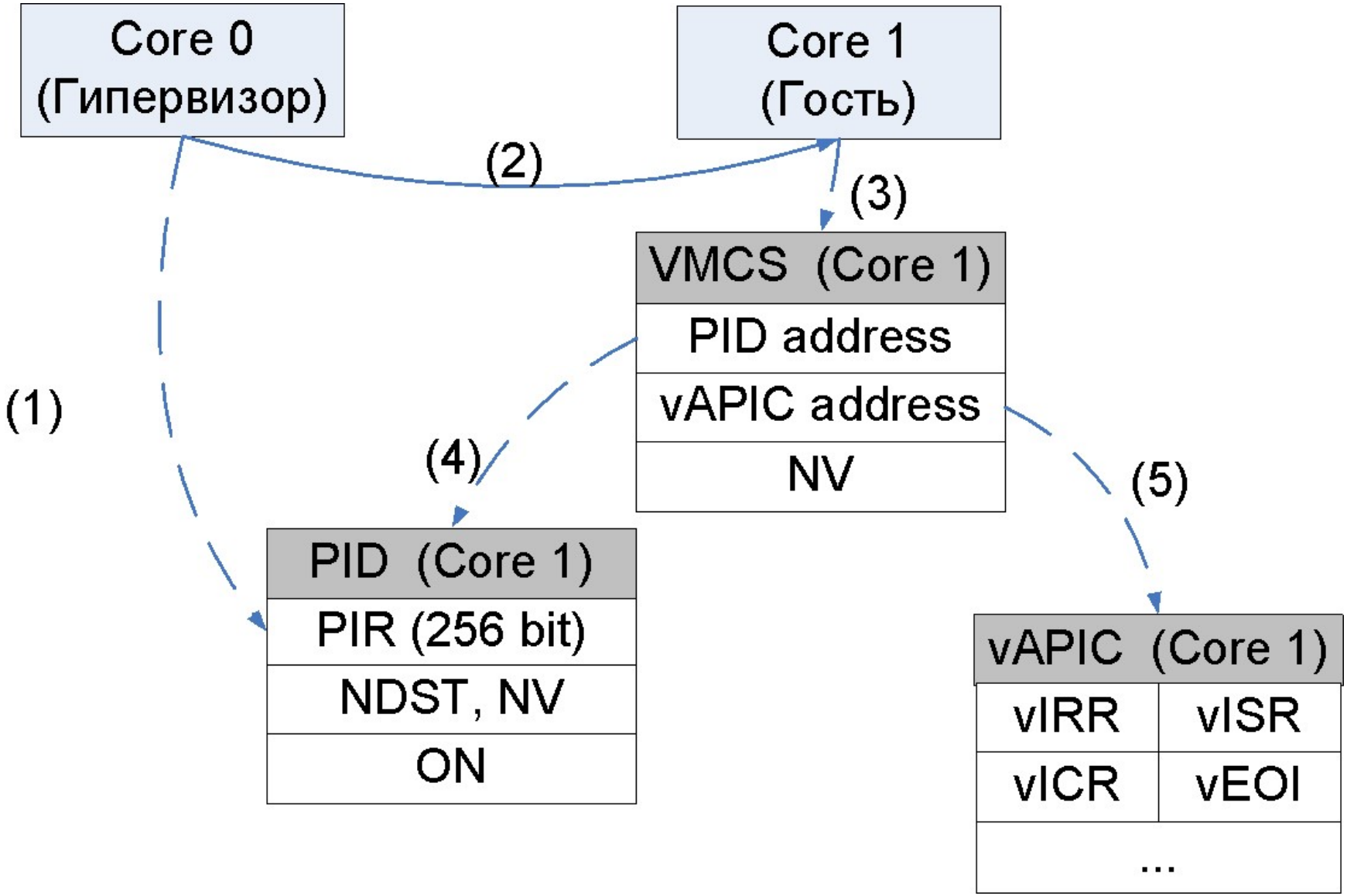
Гипервизор обращается в Core1 PID

- выставляет флаг прерывания в PIR
- читает NV, NDST, ON
- если (ON == 0), выдает прерывание ядру NDST с вектором NV

Аппаратура Core1

- определяет вектор прерывания
- сравнивает вектор со значением в VMCS
- подготавливает VMCS и vAPIC

Инжектирование прерываний: Posted Interrupts



Проброс внешнего устройства
Interrupt remapping

Трансляция в IOMMU.

(bus,dev,func) -> {guest vect, PID address}.

Далее доставка как posted interrupt:

запись в PID, отправка notification vector.

Резюме: виртуальные прерывания в Intel VT-d

- vAPIC backing page
- APIC-write emulation (vIRR, vISR, vEOI, vICR)
- Доставка прерываний через Posted Interrupt Descriptor (промежуточное хранилище). Работа через атомарные операции.
- PID содержит физ. номер ядра, вектор для уведомления (PID.nv, PID.ndst)
- Получив notification interrupt, аппаратура подготавливает vAPIC backing page и VMCS, не прерывая исполнение гостя.
- Прерывания от проброшенного устройства: Interrupt remapping в IOMMU: (bus,dev,func) -> guest vector, PID
- Работа с памятью на фоне исполнения гостя
- В x2APIC доступ к регистрам контроллера прерываний через спец. инструкции RDMSR/WRMSR