ФРТК

iSCALARE

intel

Лаборатория суперкомпьютерных технологий для биомедицины, фармакологии и малоразмерных структур

Моделирование архитектурного состояния
Моделирование доступов к памяти

Григорий Речистов

grigory.rechistov@phystech.edu

25.02.2013

На прошлой лекции

Модель ЦПУ на основе интерпретации

На этой лекции

- Моделирование арх. состояния ЦПУ
- Моделирование доступов в ОЗУ (функциональных)
- ~~Моделирование кэшей, TLB~~

Архитектурное состояние

Где хранить гостевые регистры?

- В памяти: **`uint32 registers[NUM_REGS];`**
  - Медленный доступ
- На хозяйских регистрах
  - Всё может не уместиться (IA-32 → IA-32)
- Специальный смысл некоторых регистров
- Потеря переносимости кода

Архитектурное состояние

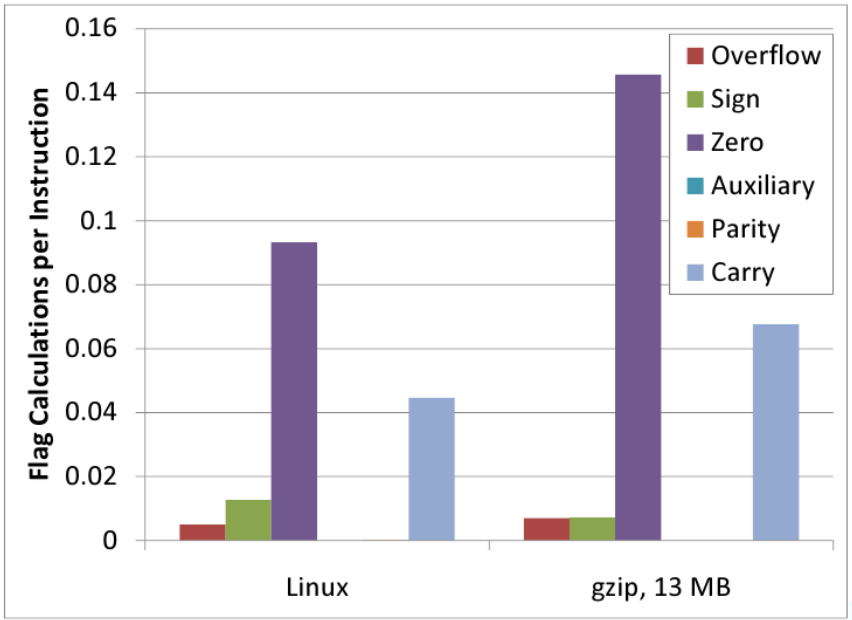
Компромисс: разместить самые часто используемые на регистрах, остальные – в памяти

- **`$PC, $SP`**
- Векторные регистры (XMM, YMM)
- **`register processor_t* cpu __asm__("rbp");`**

Архитектурное состояние

Или вообще их не хранить (не вычислять)!

- Ленивое вычисление флагов IA-32
- Не обновлять симулируемый EFLAGS, пока он не будет действительно нужен



Yair Lifshitz et al. **Zsim: A Fast Architectural Simulator for ISA Design-Space Exploration**

Доступы в память

- **`RdData = Memory[PhysAddr]; // ???`**
- **`Memory[PhysAddr] = WrData; // ???`**
- Большой объём гостевой памяти — как её хранить целиком в гостевой?
- Куда может попасть такой доступ?
  - В ОЗУ — всё ОК, обычный массив
  - В ПЗУ — запись невозможна
  - В зеркальный регион — изменения проявятся где-то в другом месте
  - В периферийное устройство (MMIO)

Ленивое выделение памяти

- Хранить только те страницы памяти, которые действительно были запрошены

```
uint8_t *hptr = hptr_lookup(guest_paddr);

if (!hptr) hptr = new_hptr(guest_paddr);

RdData = hptr[OFFSET(guest_paddr)];
```

- Необходимо учитывать ситуацию пересечения границы страниц (cross page)

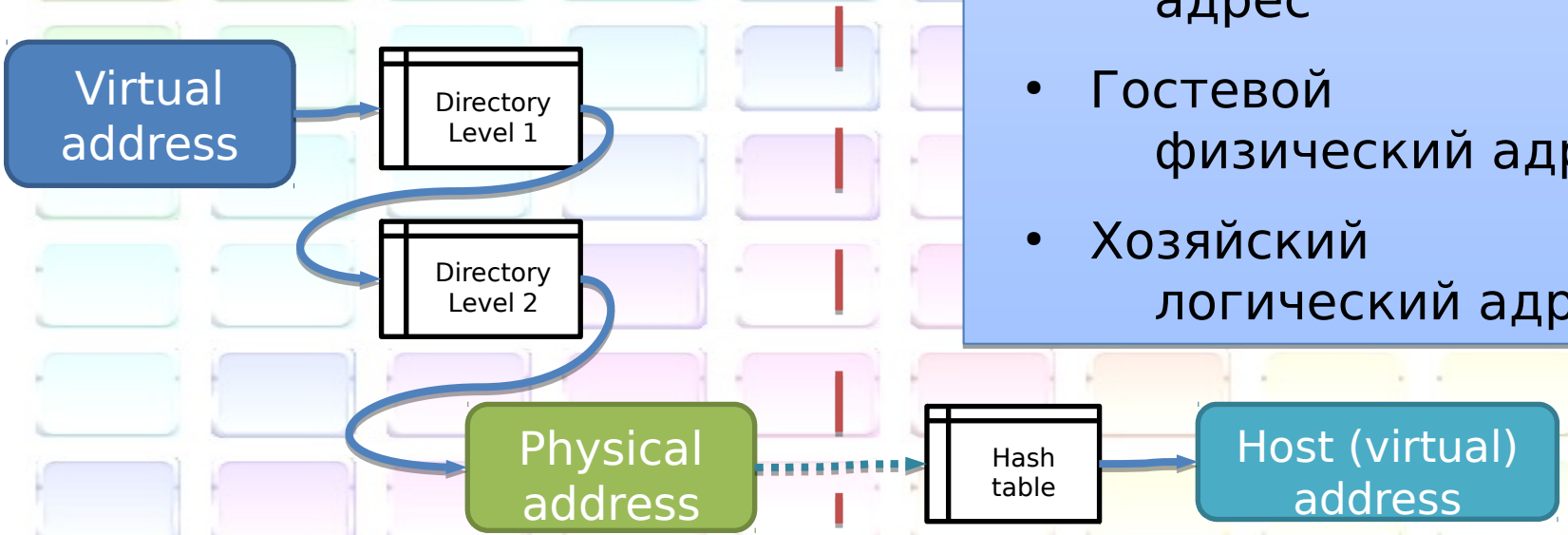
Преобразование адресов (1/2)

Аппаратура

- Виртуальный адрес
- Физический адрес

Модель

- Гостевой виртуальный адрес
- Гостевой физический адрес
- Хозяйский логический адрес



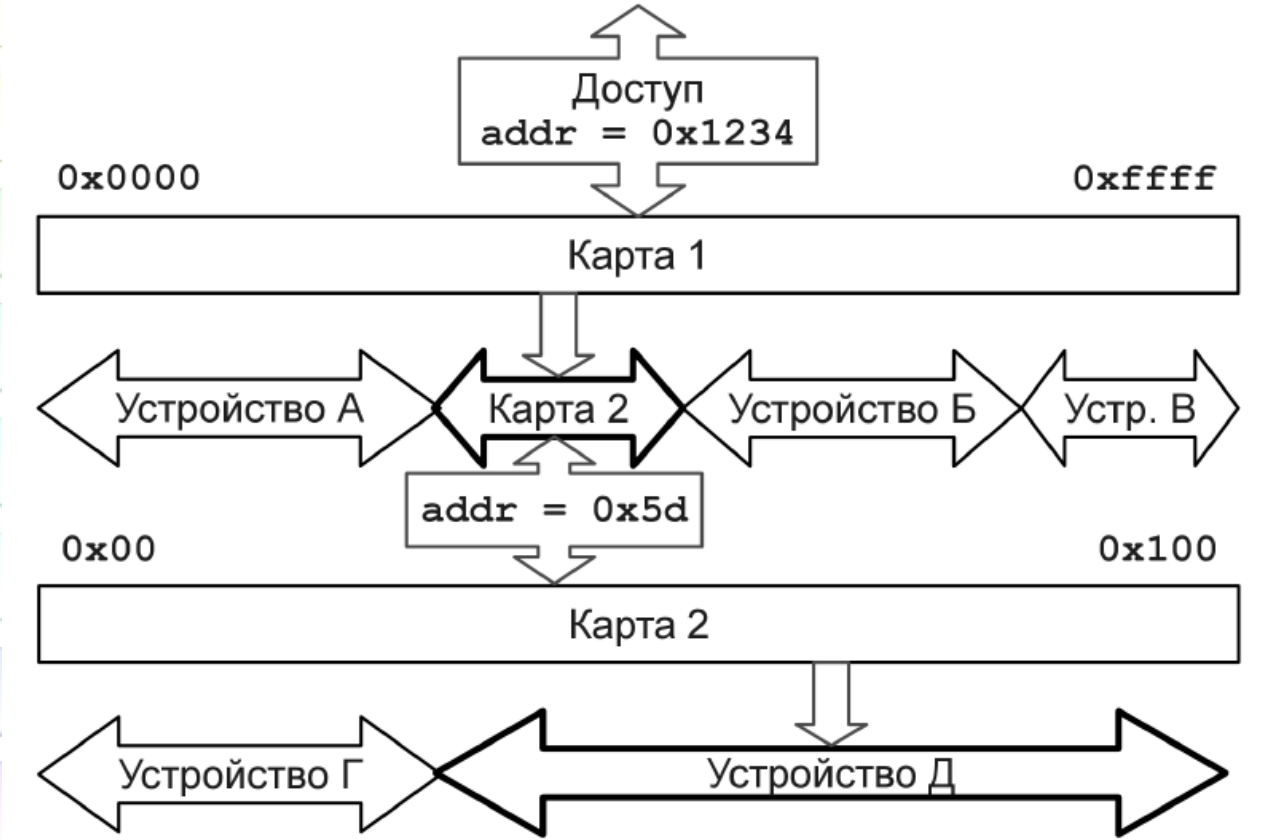
Преобразование адресов (2/2)

- Преобразование **v2p** требует нескольких чтений памяти/регистров – медленно.
- Этого пытаются избежать с помощью устройств TLB (translation look aside buffer)
  - наиболее часто используемые пары адресов хранятся в TLB, время доступа к ним – 1 такт.
- Аналогичный подход в симуляции – таблица **v2h** (softTLB)

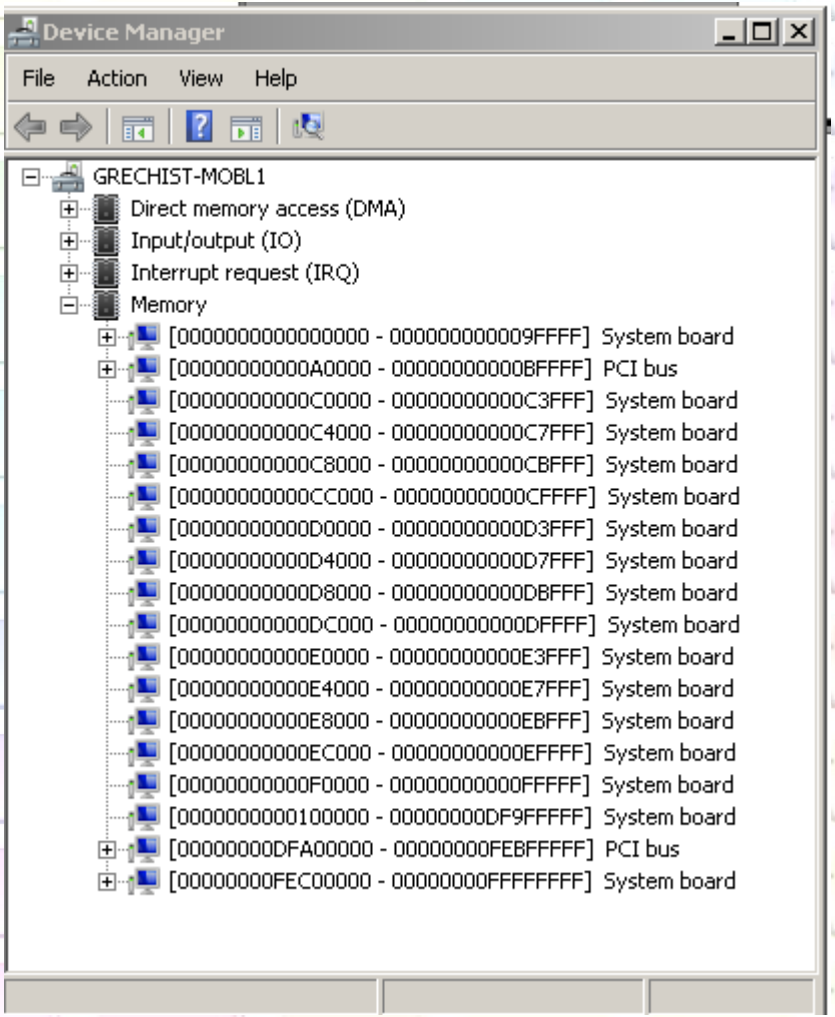
Моделирование MMIO

- Типы доступа: read, write, fetch, prefetch, *inquiry*
- Интерфейс периферийных устройств
  - **`read(offset)`** → **`uint64 result/exception`** + побочные эффекты
  - **`write(offset, value)`** → **`exception`** + побочные эффекты
  - **`read8, read16, read32, read64,`** …
  - **`write8, write16, write32, write64,`** …

Моделирование MMIO — карты памяти



Карты памяти



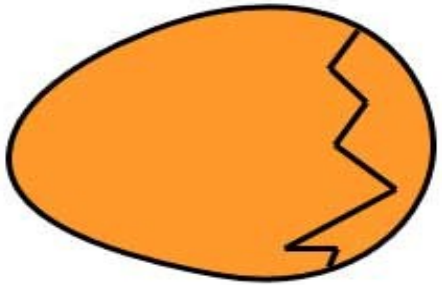
Порядок байт при доступах

- Big Endian
- Little Endian

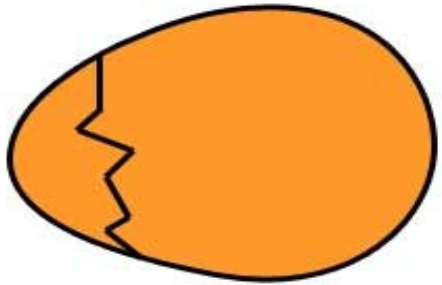
Представление		D4*0x01 + C3*0x100 + B2*0x10000 + A1*0x1000000
Порядок от младшего к старшему	(little-endian)	0xD4, 0xC3, 0xB2, 0xA1
Порядок от старшего к младшему	(big-endian)	0xA1, 0xB2, 0xC3, 0xD4

- Mixed Endian

Endianness



BIG ENDIAN - The way people always broke their eggs in the Lilliput land



LITTLE ENDIAN - The way the king then ordered the people to break their eggs

Байт, слово

- Бит — ?
- Байт — ?
- Машинное слово — ?

Байт, слово

- Байт — минимальная адресуемая (в данной архитектуре) единица хранения информации
- Октет — восемь бит
- Машинное слово — максимальный объём информации, который ЦПУ может обработать единовременно
- Intel: word — 16 бит, dword — 32 бит, qword — 64 бит

Ресурсы для дополнительного чтения

- University of Kansas. EECS 700 Virtual Machines
  http://www.ittc.ku.edu/~kulkarni/teaching/EECS700-VM/index.html
- Stanislav Shwartsman, Darek Mihoka. **How Bochs Works Under the Hood. 2nd edition.**
  http://bochs.sourceforge.net/HowtheBochsworksunderthehood2ndedition.pdf
- M. Domeika, M. Loenko, P. Ozhdikhin, E. Brevnov. **Bi-Endian Compiler: A Robust and High Performance Approach for Migrating Byte Order Sensitive Applications**
  http://cerc.wvu.edu/download/WORLDCOMP'11/2011%20CD%20papers/ESA2902.pdf

На следующей лекции:

- Можно ли сделать симулятор, работающий быстрее, чем интерпретатор?
- Двоичная трансляция

Спасибо за внимание!

Все материалы курса выкладываются на сайте лаборатории:
http://iscalare.mipt.ru/material/course_materials/