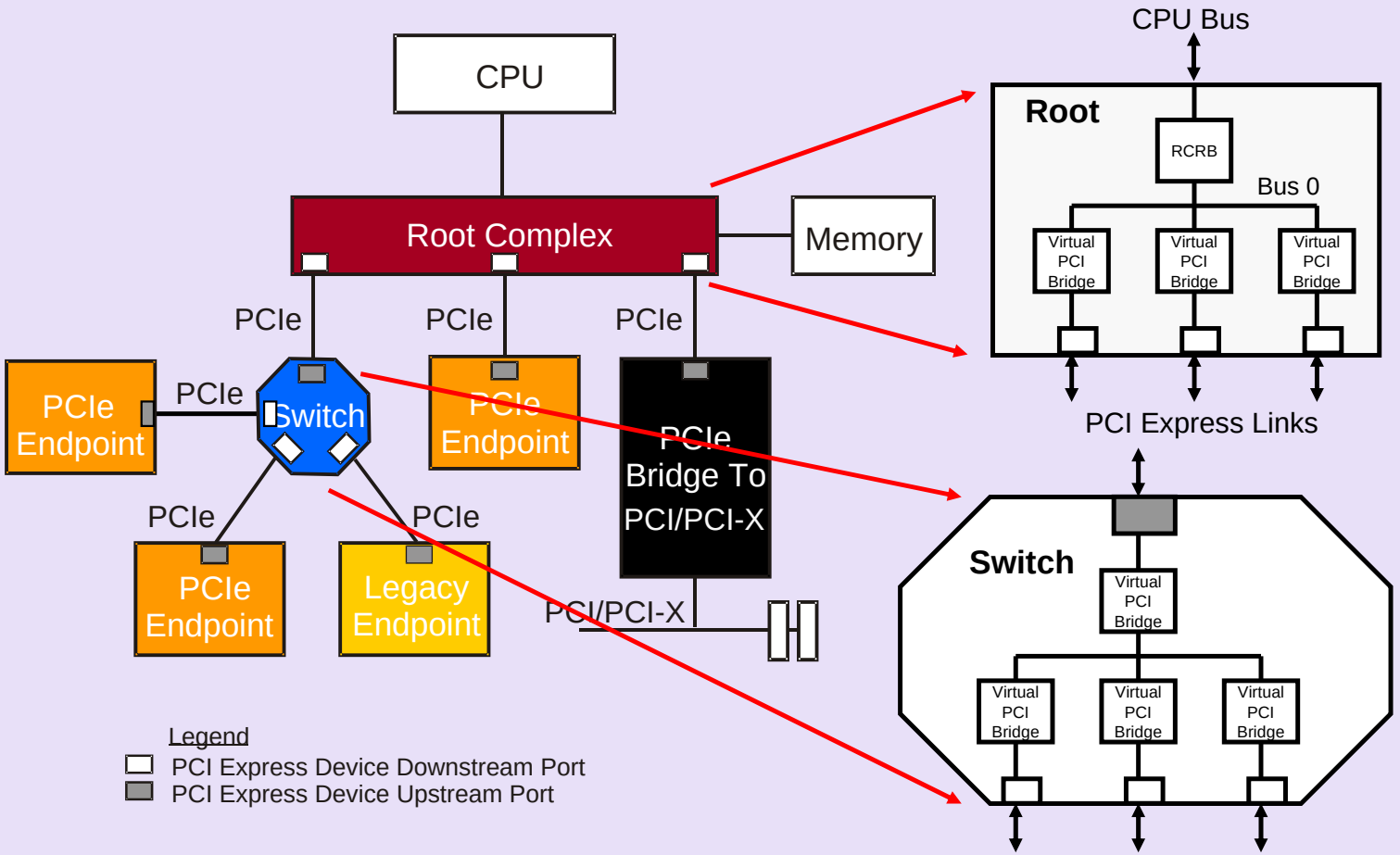
VT-d and FreeBSD

Константин Белоусов
kib@freebsd.org

21 сентября 2013 г.

Revision : 1.11

Example PCI Express Topology – Root & Switch



TLP - Transaction Layer Packets

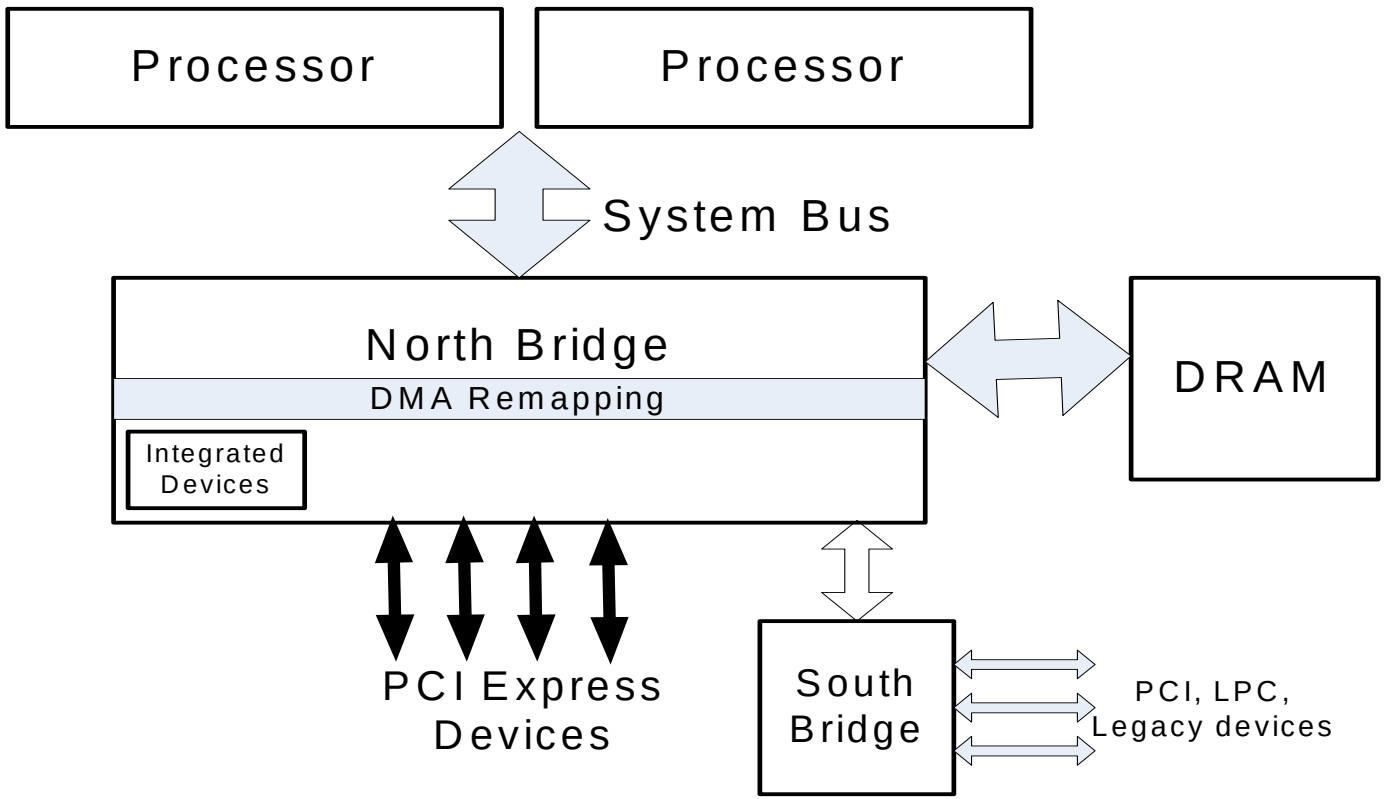
- I/O
  - Host access to device (BARs)
  - Device access to memory (DMA)
  - Peer to peer
    - GPU RDMA over Infiniband
    - Nvidia Optimus
- Messaging: Interrupts, Errors
- Configuration I/O.

Device DMA engines

Features and Limitations

- Scatter/Gather: number of segments
- DMA engine restrictions
  - Address width
  - Dead bits (alignment)
  - Segment length
- Streaming
- Coherence (Snoop)
- Traffic Prioritization

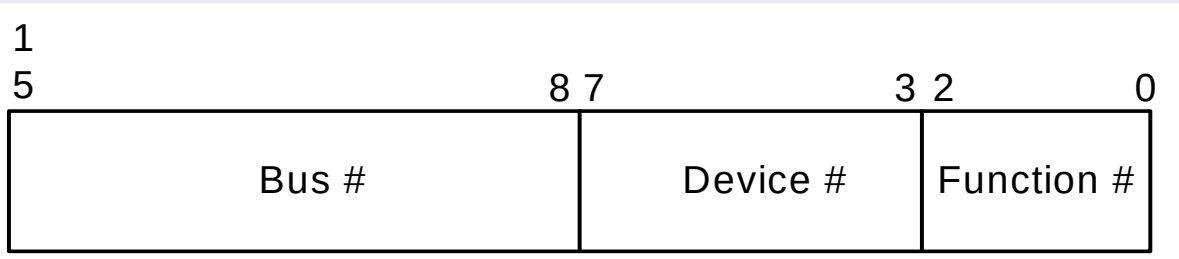
VT-d


Processor
Processor
System Bus
North Bridge
DMA Remapping
Integrated Devices
DRAM
PCI Express Devices
South Bridge
PCI, LPC, Legacy devices

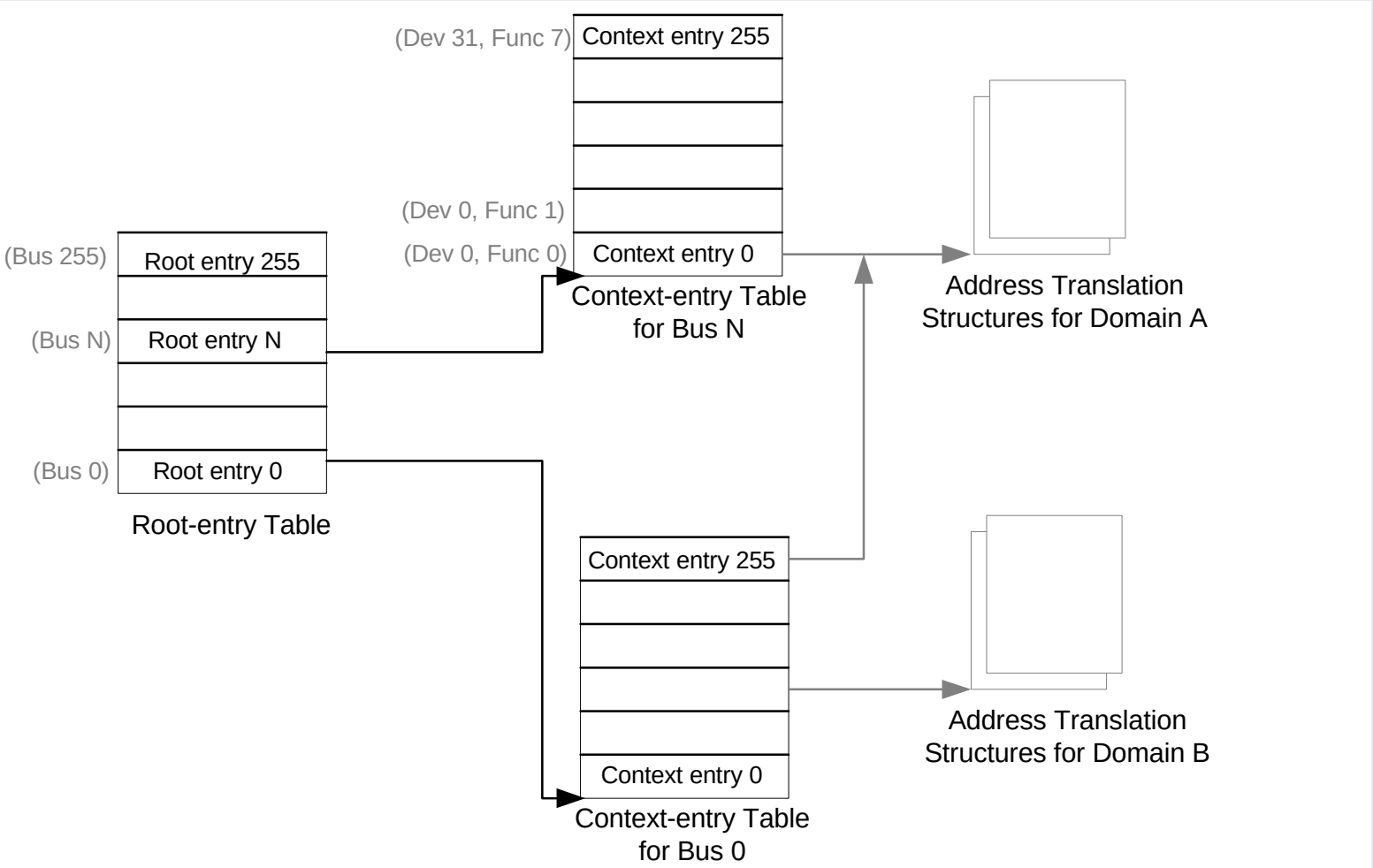
DMAR

- Process TLPs from devices accessing memory
- Performs
  - Address Translation and Access Control
  - Snoop Control
  - Prioritization
- Based on the originator of the TLP

Requester Identifier



DMAR translation structures



VT-d

Hardware

- Nehalem+ Xeons
- Desktop Core i7 CPUs: not -K, BIOS
- Core2 gen: G45, 5500

Documentation

- Intel® Virtualization Technology for Directed I/O, **D51397-005**
- *External Design Specification* (EDS)
- *BIOS Write Guide* (BWG)
- Chipset erratas

VT-d

Compatibility

- SMI handlers, USB legacy
- UMA GPU: GTT and VGA framebuffer
- Service processor for BMC (AMT, IPMI, iLO, DRAC etc)

Bugs

- Hardware bugs, Specification Updates
- BIOS bugs

How to detect

acpidump -t

```
DMAR: Length=368, Revision=1, Checksum=7,
    OEMID=DELL, OEM Table ID=PE_SC3, OEM Revision=0x1,
    Creator ID=DELL, Creator Revision=0x1
    Host Address Width=46
    Flags={INTR_REMAP,X2APIC_OPT_OUT}
```

Other features

- Interrupt remapping
  - MSI, MSI-X: memory write
  - IO-APICs
  - FSB interrupts: HPET
- ATS (Address Translation Service): IO TLB in devices
- Hypervisors PCI pass-through

PCI-era

- Architectures
  - SPARC4u
  - POWER: DART
- coarse domains

DMA in FreeBSD

Busdma(9) layer

- FreeBSD KPI abstracting access to DMA implementations
- from NetBSD

Busdma(9) overview

- Tags: device capabilities
- Maps: Accessible memory
- Loads and unloads: maps activation and deactivation

Busdma implementations

Bounce buffers

- Allocate memory to satisfy device constraints
  - contigmalloc(9)
  - Low 16MB, low 4GB
- Copy to/from
- Flush cache on non-coherent platforms

Busdma implementations

IOMMU: pro

- Performance: No bouncing
- Stability: No memory corruption
- Privacy: Only sanctioned access to memory
- Driver debugging: Reports of violations

IOMMU: contra

- Performance: Page table setup
- Performance: Translation overhead

Busdma over VT-d

Layers

- Page tables and TLB invalidation
- Fault handler
- Context and domain
- Busdma emulation

Integration

- ACPI: DMAR table parsing
  - DMAR discovery
  - RMRR and BIOS bugs
- newbus: `bus_get_dma_tag()`
- fallback to bounce, enabling pass-through

Busdma over VT-d

Busdma KPI problems

- Locking
  - `BUS_DMA_NOWAIT` abuse
  - `bus_dmamap_unload(9)` cannot sleep
- No I/O direction
- Tag specification of alignment

Busdma over VT-d

Current state

- Drivers
  - Storage: ahci(4), mfi(4)
  - USB: uhci(4), ehci(4)
  - Network: em(4), igb(4) (*), bce(4)
- Platforms
  - Xeon 5400, 5500 NB
  - Xeon Romely-EP (E5-26XX)
  - Haswell (Core i7 4770)
- Not supported yet
  - Intel GPUs
- Not tested
  - HDA
  - Discrete GPUs (Radeon, Nvidia)
  - Everything else (HW bugs)