PCI Express Support in QEmu

Isaku Yamahata <yamahata@private.email.ne.jp>
<yamahata@valinux.co.jp>
VA Linux Systems Japan K.K.
LinuxConJapan 2010: September 29, 2010

Agenda

- Introduction
- Current status and implementation
- Example
- Future work
- Summary

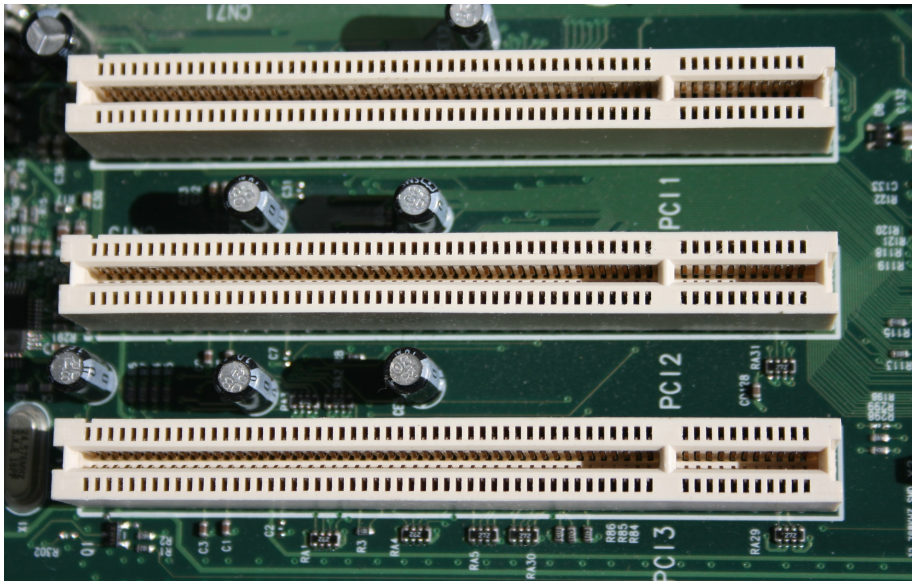
Introduction

Motivation

- QEmu is used for device emulator for many virtualization technologies. KVM, Xen...
- QEmu supports PCI in a limited way, and doesn't support PCI Express.
- So do QEmu derivatives.
- Fill those gaps
  - Address them to enable KVM, Xen, ... to utilize those features.

What's PCI?

- Peripheral Component Interconnect
- Year created:1992
- Parallel bus
- Has been widely adopted in the market



From Wikipedia

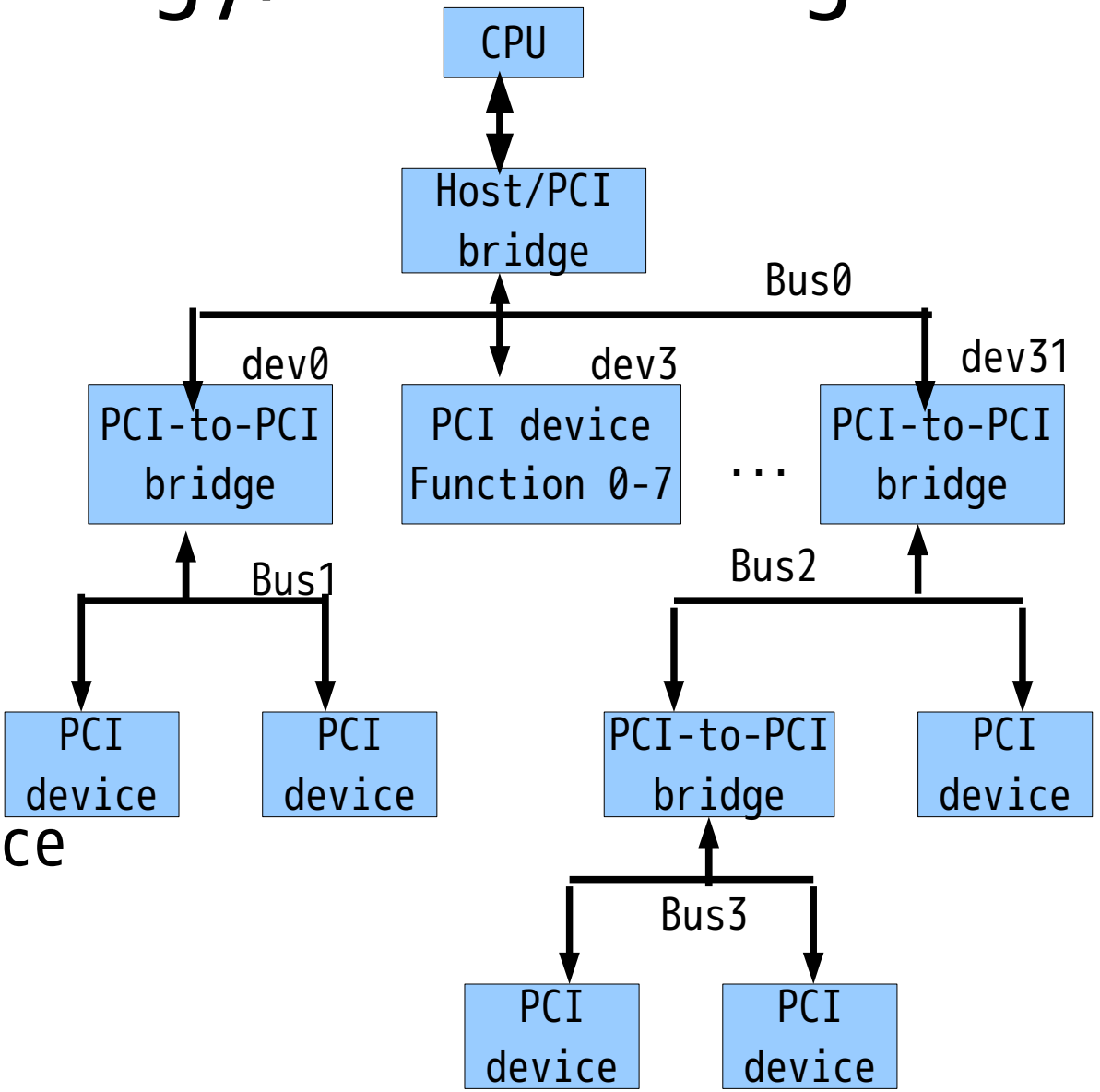
PCI features from software point of view

- Bus topology/addressing
- Configuration space
- BAR(Base Address Register)
- Interrupt

From wikipedia

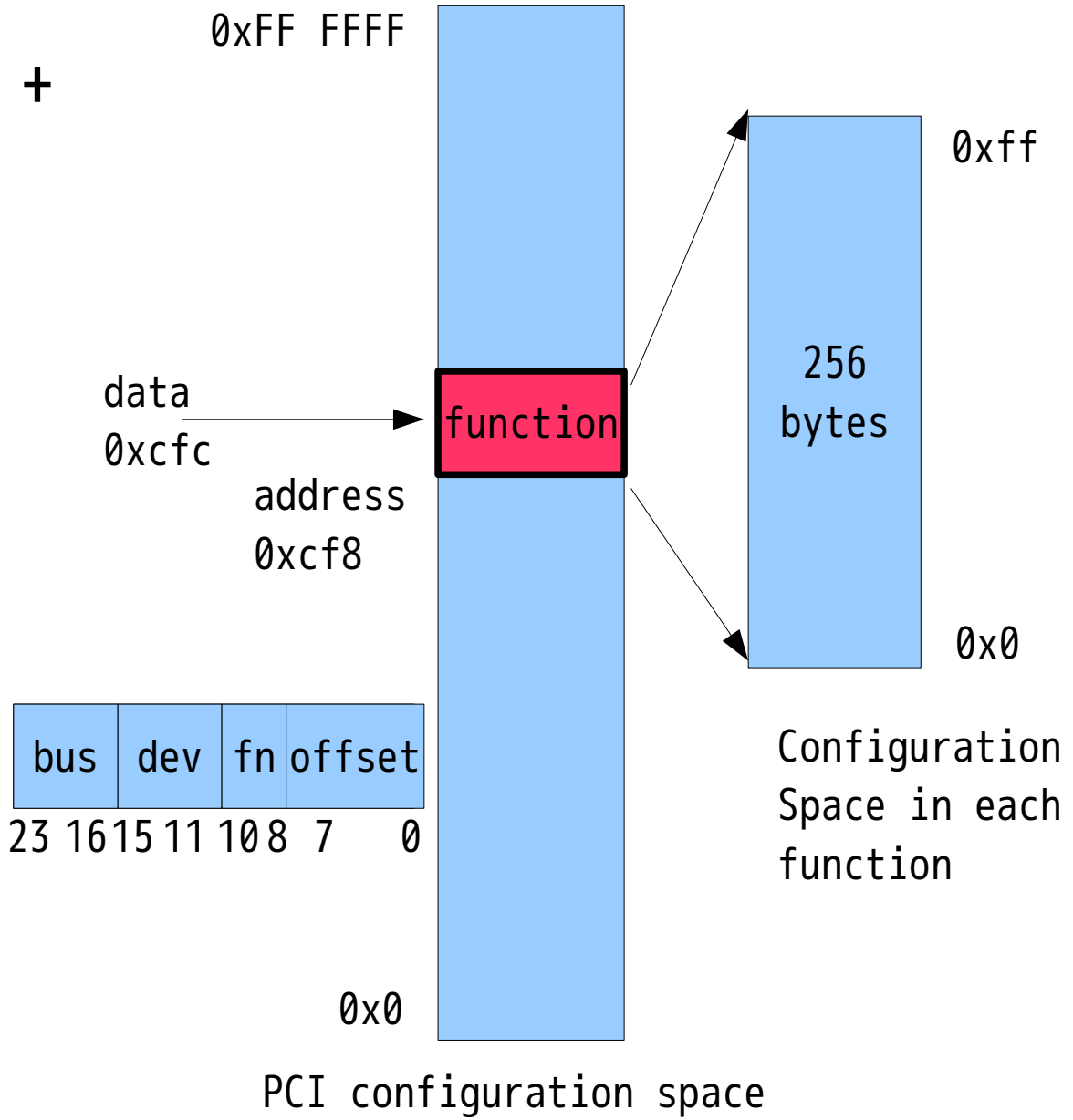
PCI bus topology/addressing

- Bus addressing: 3 addressing spaces
  - Memory: accessed via MMIO
  - IO: accessed via IOIO
  - Configuration space



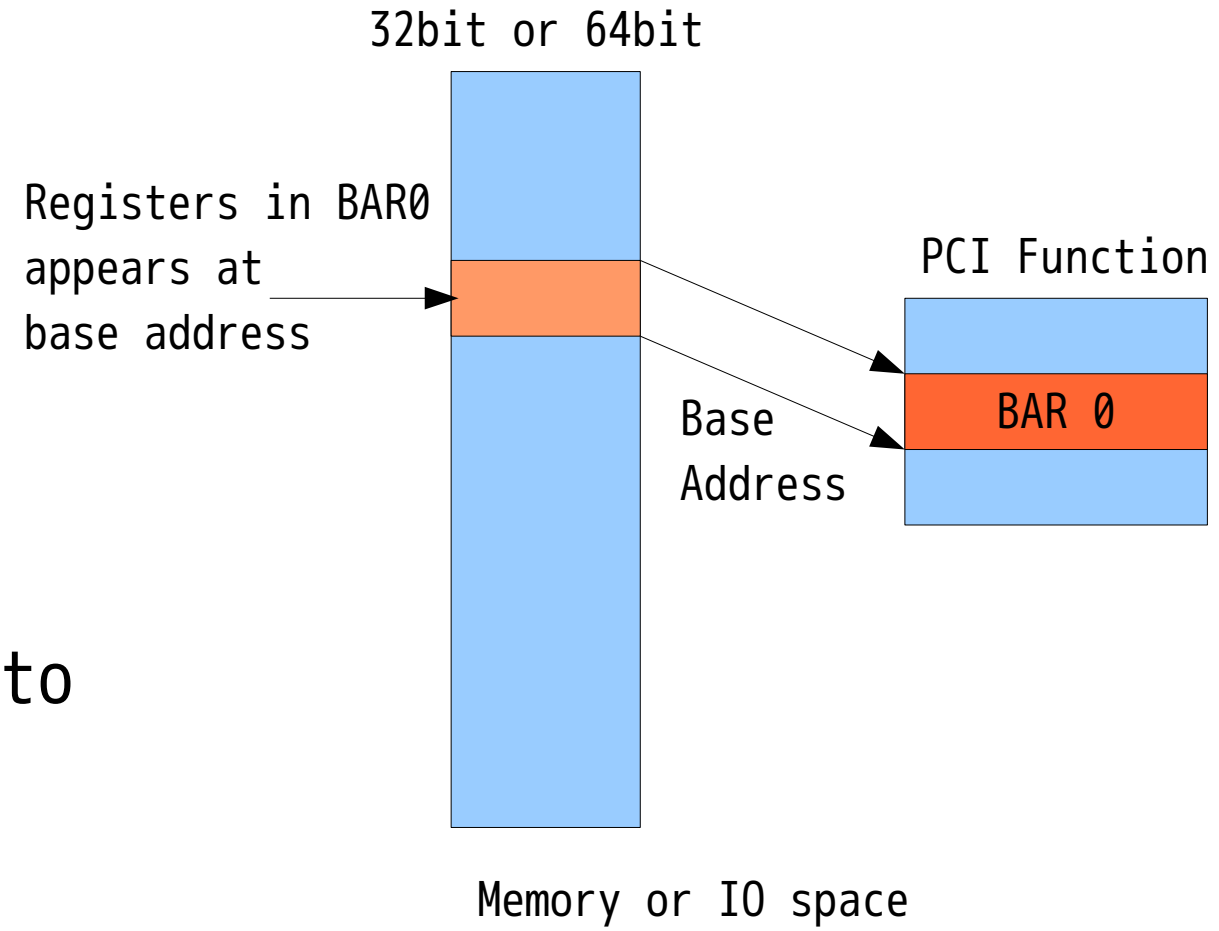
PCI configuration space

- Bus,device,function + offset
- 256 bytes on each function
- Indirect access via IO port
  - 0xcf8: address to configuration space
  - 0xcfc: data



BAR(Base Address Register)

- Memory
  - 32bit/64bit
- IO
  - 32bit
  - x86 is able to access only up to 16bit.

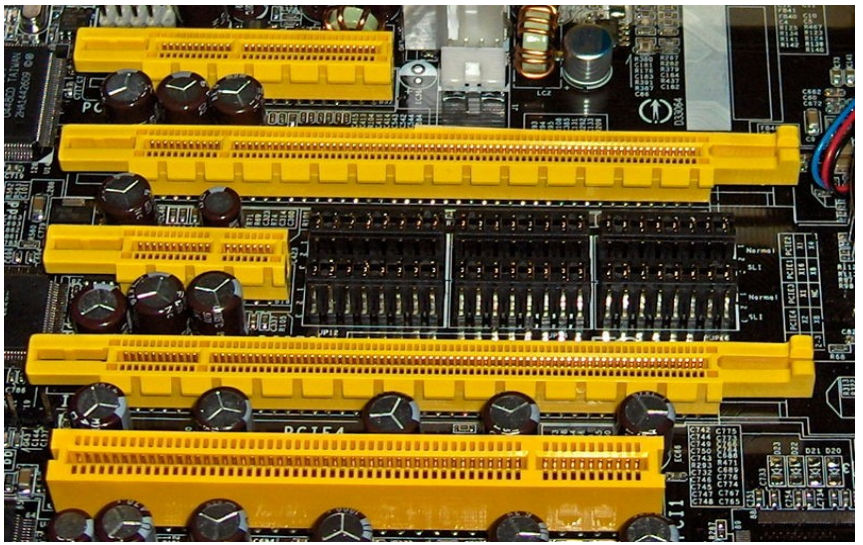


Interrupt

- INTx#
  - 4 interrupt lines per device
    - INT[A-D]#
  - edge/level triggered
  - Interrupt routing table in BIOS, ACPI
- MSI/MSI-X: Message Signaled Interrupts
  - Memory write
  - No routing issue

What's PCI Express?

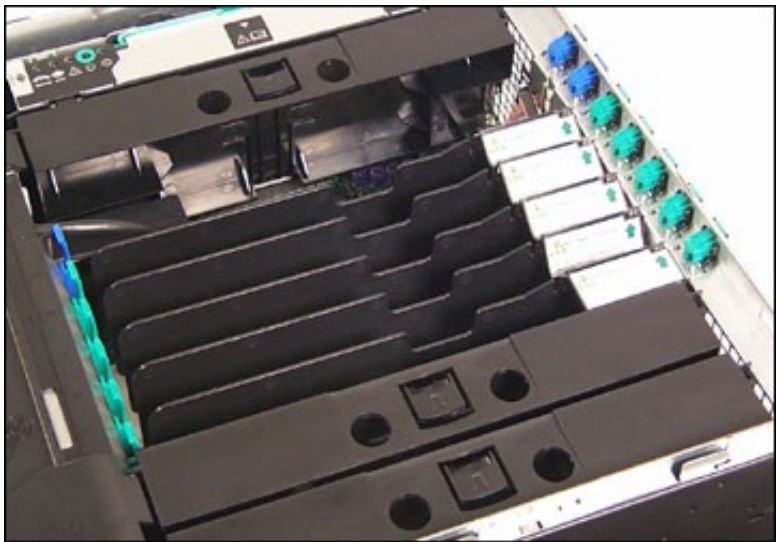
- Designed as a successor of PCI
  - Software compatible with PCI
  - Many improvements
  - Widely accepted in the market
  - Has been superseding PCI
- Year created: 2004
- Serial bus



From Wikipedia

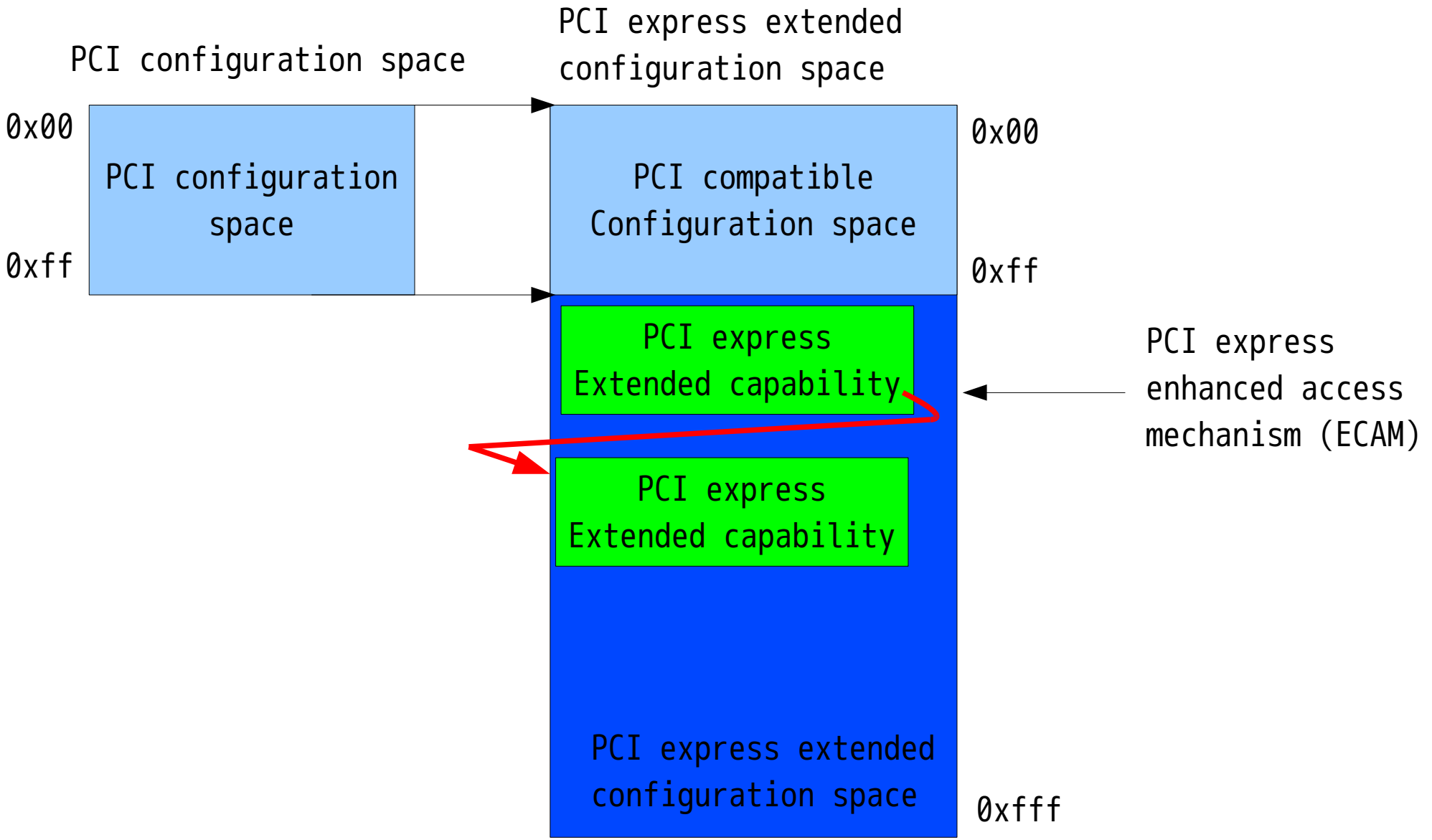
Express features from software point of view

- Many enhancements from PCI, for example
  - MMCONFIG: larger configuration space
  - Native hotplug:not ACPI based
  - Native power management
  - AER(Advanced Error Reporting)
  - ARI(Alternative Routing ID)
  - VC(Virtual Channel)
  - FLR(Function Level Reset)

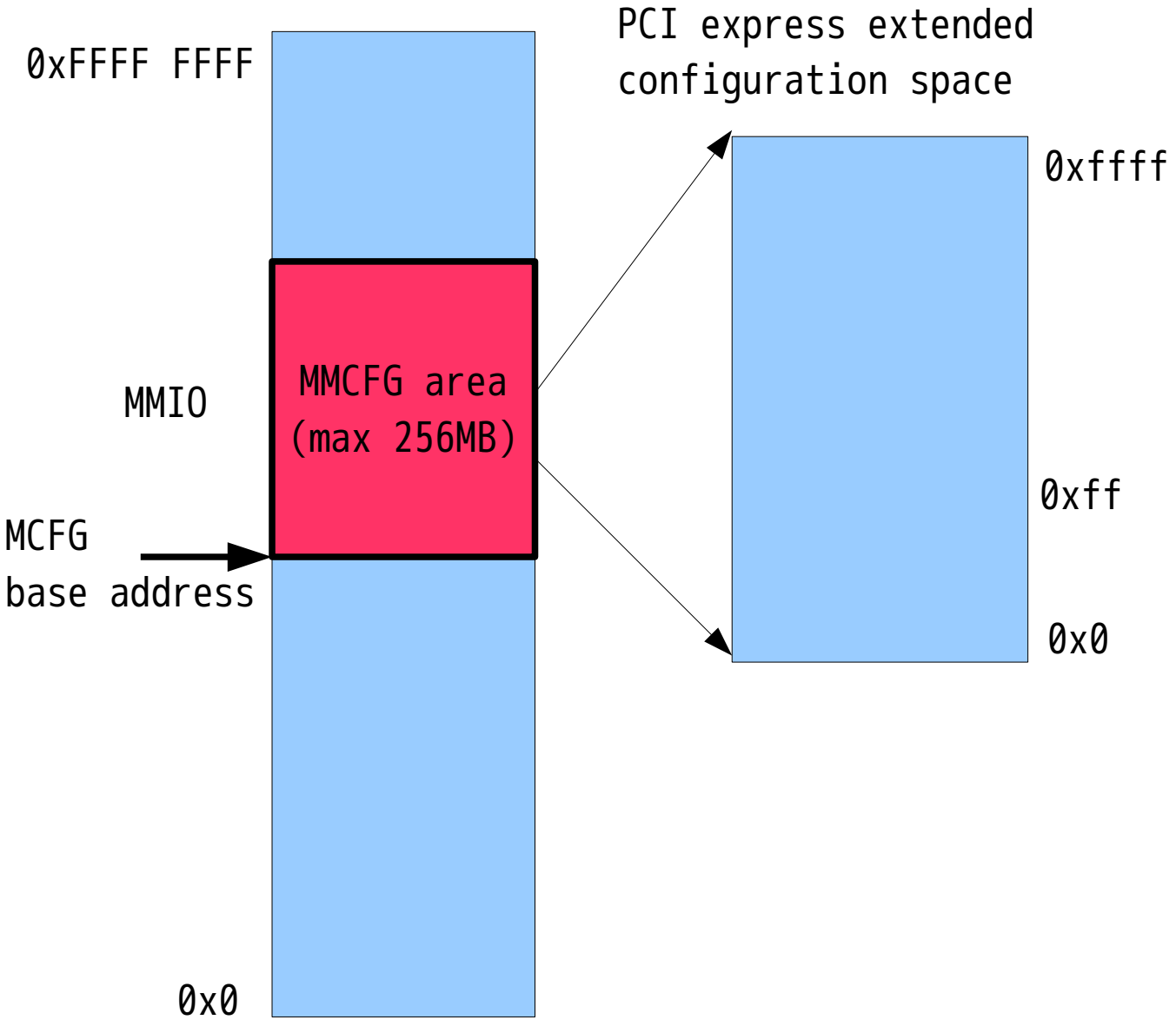


From http://cdnsupport.gateway.com/s/Servers/9715Server/54.jpg

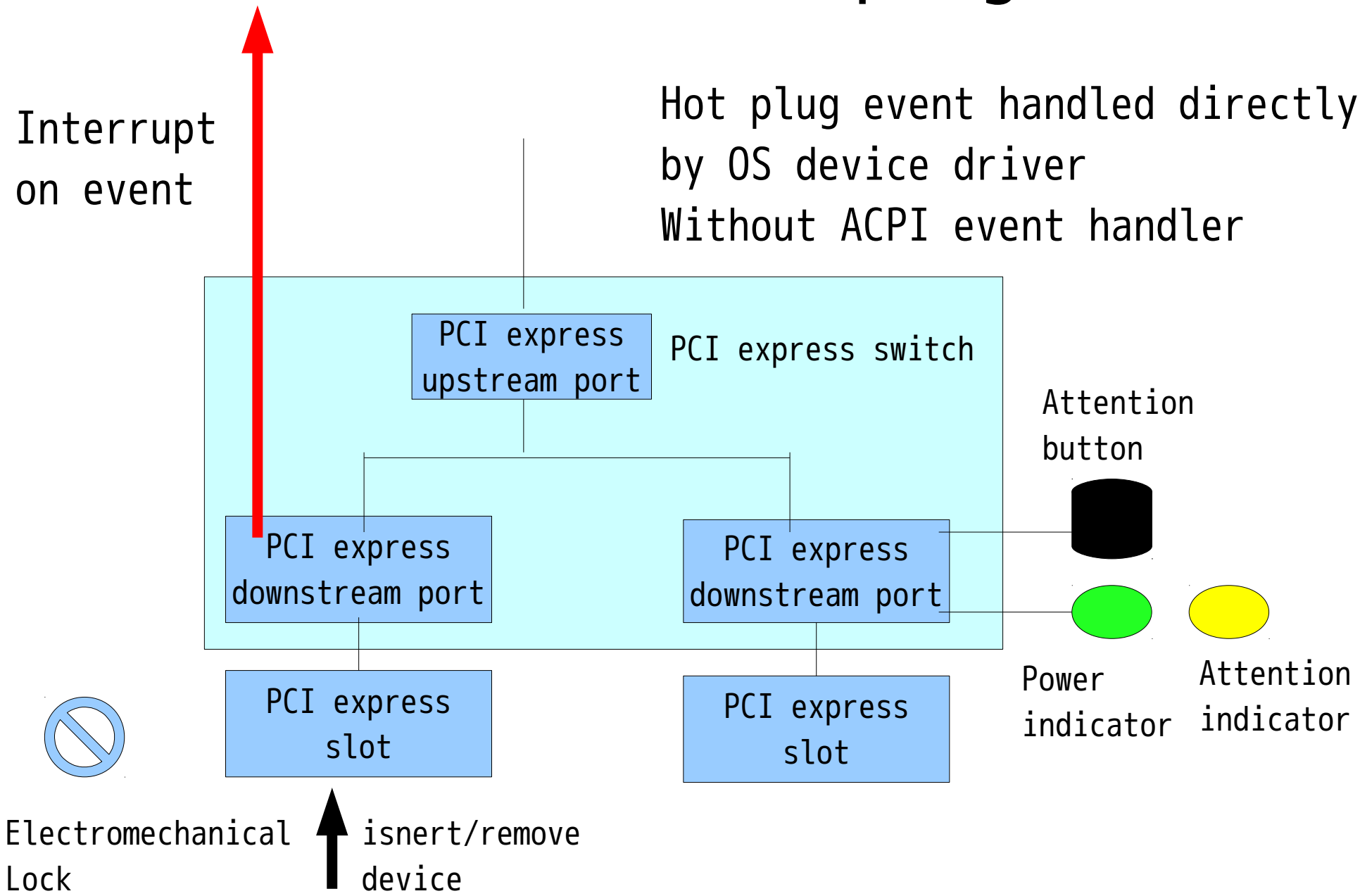
PCI express extended configuration space



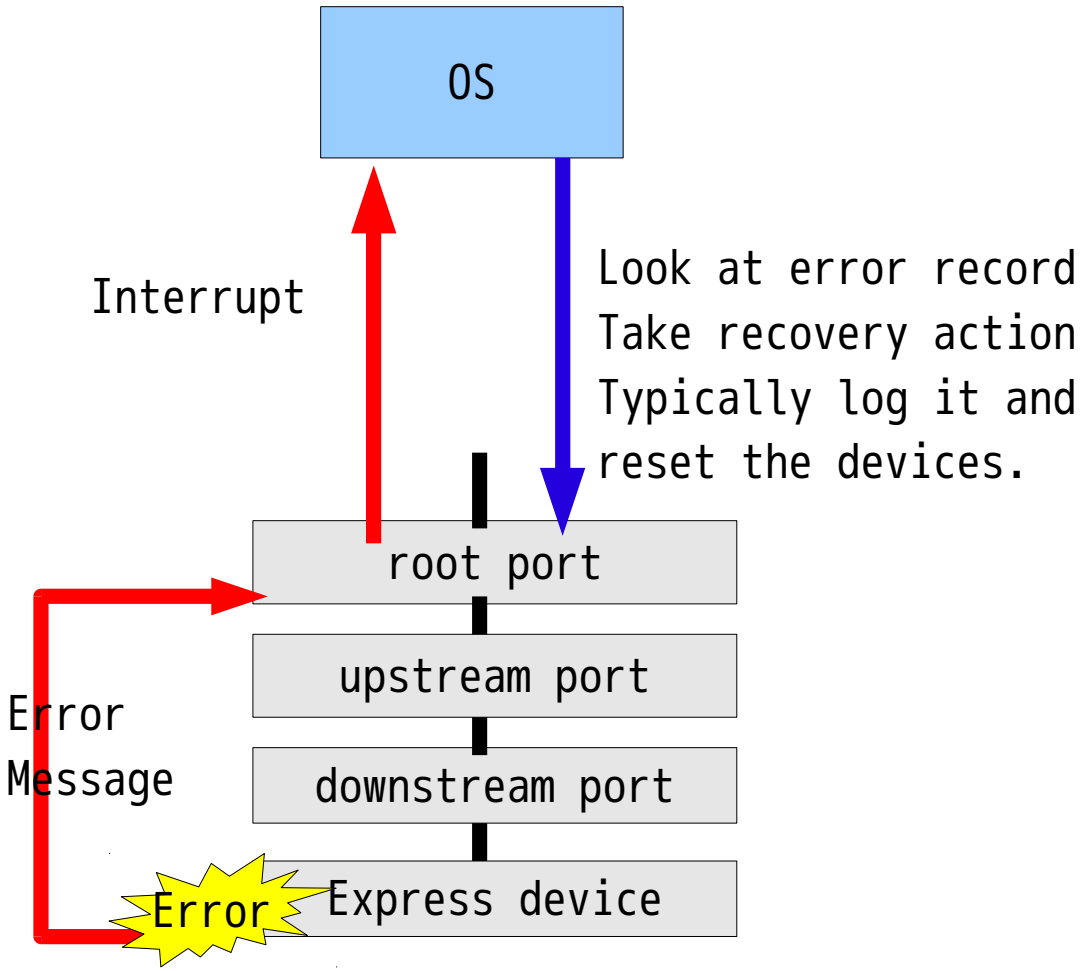
PCIe MMCONFIG



Native hot plug



Advanced Error Reporting(AER)



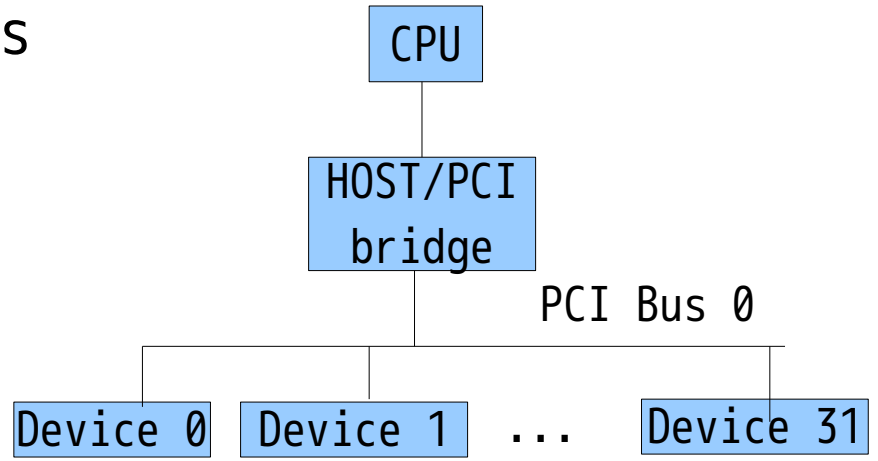
- Standardized error reporting.
- Important for RAS

Why PCI Express?
Isn't it compatible with PCI?

- Upper compatible
- Many new native features
  - They can be only used via express feature.
- Some device drivers require native express
  - They check if the device is really express
  - Existing PCI device assignment doesn't suffice
- Hardware certification requires express

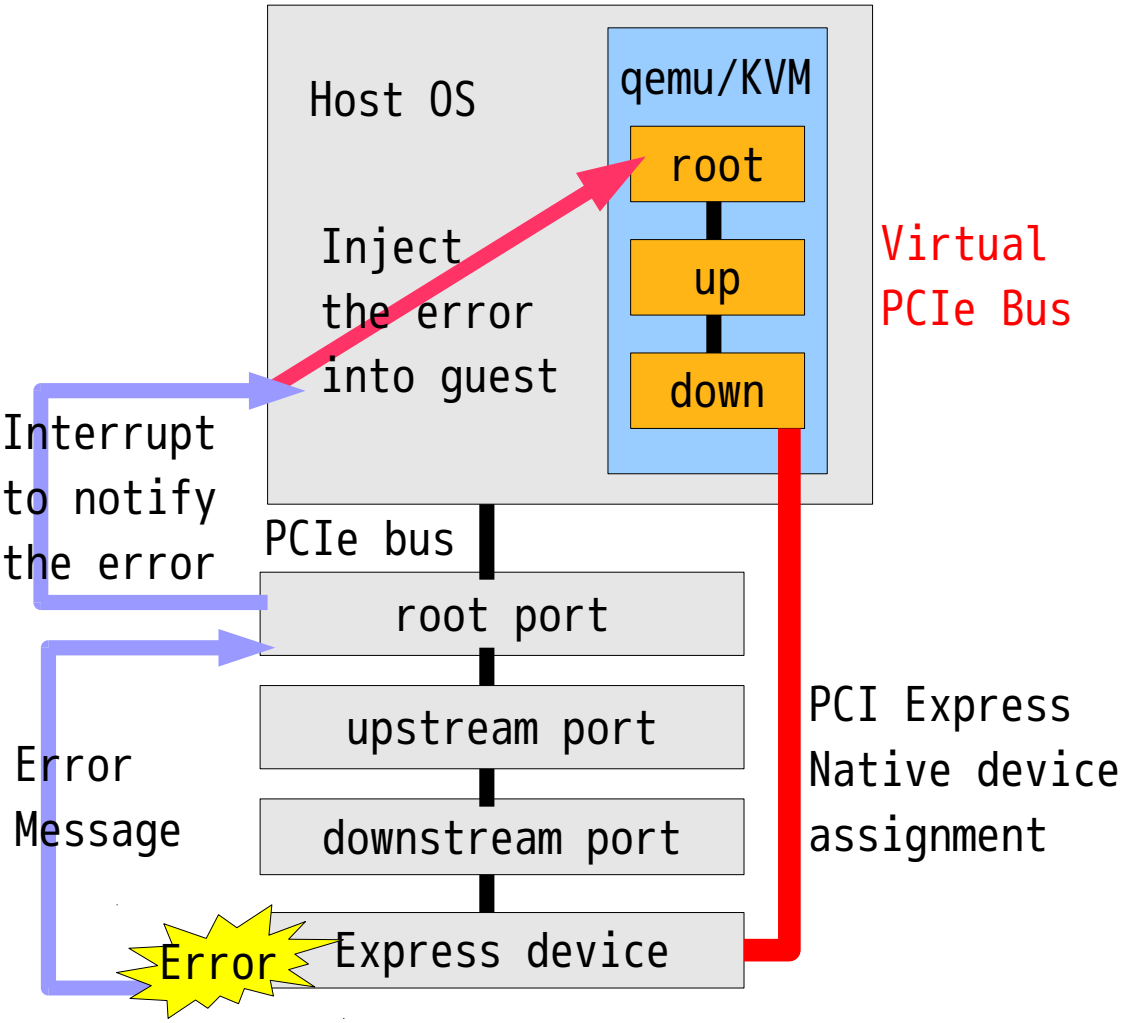
Goal in PCI area

- Enable 3+ pci buses(96+ slots)/96+ pcie slots
  - The current PC emulation supports only host bus.
    - Flat PCI topology: up to only 32 devices
  - PCI hotplug requires ACPI dance.
    - The used DSDT supports only pci bus 0.
    - This is difficult to resolve with acpi
- Enable unsupported features
  - 64bit BAR
  - Multifunction bit
  - Bridge filtering
  - ...



Goal in PCI Express area

- Enable QEmu to support PCI Express
- Enable PCI Express native device assignment with
  - Native hot plug
  - AER(RAS)
- Then, bring Express support to qemu derivatives.



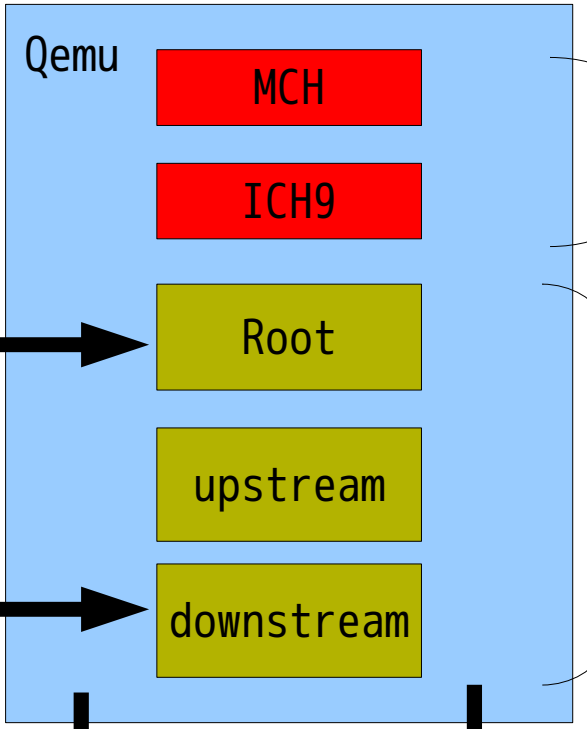
Current status and implementation

Merged
Under review
To be posted

I440fx chipset refactoring
64bit BAR
Extended config space
MMConfig
PCI-to-PCI bridge clean up
PCI bus reset

AER error injection
pcie_aer_inject_inject

Native hotplug
pcie_abp



Q35 chipset
New DSDT

PCI express port switch

Pass DSDT
(avoid rom
size limit)

PV pci bus numbering
Pass hint for pci bus number

chipset abstraction(i440fx)
64bit BAR
Multi pci bus init
DSDT loading
MCFG
Q35 support

Seabios

Hot plug function

Function	Supported?
Attention Button	yes
Power Controller	No
MRL Sensor	No
Attention Indicator	Yes
Power Indicator	Yes
Hot-Plug Surprise	Yes
EMI	Yes

Why new chipset?

- The current supported chipset is very old
  - For Pentium Pro/II/III
  - North bridge: I440FX
  - South bridge: PIIX3 (and PIIX4 for acpi power management and pci hot plug)
  - Hardware release date: May 1996
- Too old for new hardware features

From wikipedia

Why new chipset?(cont.)

- Add new features for modern OSes without legacy compatibility.
  - Discard legacy compatibility
  - It's very difficult to test various legacy OSes
  - Only for modern OSes
- Keep the old chipset emulator for legacy compatibility.

New chipset emulator

- Q35 chipset based
  - For Core2 Duo
  - North bridge: mch
  - South bridge: ich9
  - Release date: Sep 2007

From wikipedia

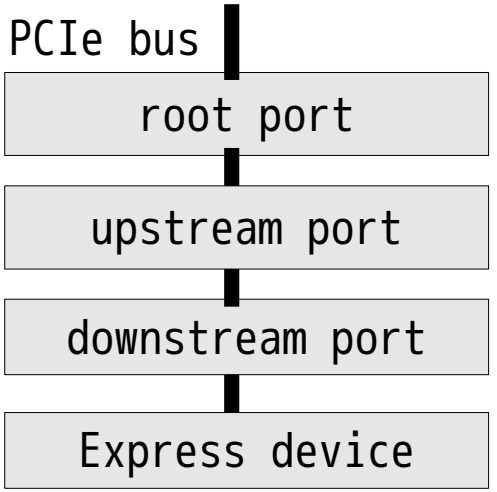
- In fact I have chosen Q35 because I have it available at hand.
  - Newer chipsets(gmch/ioh, ich10) have mostly same feature from the point of view of emulation except graphics.

Q35 chipset emulator doesn't have

- IOMMU(VT-d) emulation
  - IOMMU emulation is coming by others
    - Only for emulated devices,
    - Not for direct assigned devices.
- Integrated graphic emulation
  - So it should be called P45, not Q35?

PCI Express port emulator

- Root/upstream/downstream port
  - All of three ports are needed.
  - Necessary for native hot plug, AER.
  - Native hotplug
  - AER
- Clean up of PCI bridge
  - It was just a stub, had to implement it first.
- Bus numbering
  - Paravirtualize to allocate range of bus numbers for hot plugged pci-to-pci bridge



SeaBIOS modifications

- Multi chipset support
  - factor out i440fx specific code
- PCI Bus initialization
  - 64bit BAR
  - Multiple PCI buses
  - Bus numbering paravirtualization
- ACPI MCFG to specify MMCONFIG area
- Passing DSDT from qemu command line to guest bios

Seabios Modifications(cont.)

- E820 update
  - Make e820 code 64bit aware.
    - So far it filled higher bits with zero.
  - Linux requires MCFG area is covered by e820 reserved area
  - Otherwise Linux thinks that it's bios bug and avoids to use MMCONFIG.

Current status

QEmu

Items	Status
64bit BAR	Merged
PCI Bridge lib	Merged to PCI branch
PCI Bus reset	Under review
MMCONFIG(PCI layer)	Merged
PCIe port switch Including native hotplug AER error injection	Under review
DSDT overriding	posted(to be resend)
Q35 Chipset	To be posted
PV PCI bus numbering	To be posted

Seabios

Items	Status
64bit BAR	Merged
Multi pci bus	Merged
Chipset abstraction	Merged
DSDT overriding	Under review
MCFG	Under review
Q35	To be posted
Q35 DSDT	To be posted
PV pci bus numbering	To be posted

VGABios

Items	Status
VBE	Waiting Gerd's patch

Example

Example from Linux boot log

```
ACPI: RSDP 00000000000f7ae0 00014 (v00 BOCHS )
ACPI: RSDT 000000001ff78f90 00038 (v01 BOCHS  BXPCRSDT 00000001 BXPC 00000001)
ACPI: FACP 000000001ffffe70 00074 (v01 BOCHS  BXPCFACP 00000001 BXPC 00000001)
ACPI: DSDT 000000001ff78fd0 86C82 (v01   BXPC   BXDSDT 00000002 INTL 20100121)
ACPI: FACS 000000001ffffe00 00040
ACPI: SSDT 000000001ffffdc0 00037 (v01 BOCHS  BXPCSSDT 00000001 BXPC 00000001)
ACPI: APIC 000000001ffffce0 00072 (v01 BOCHS  BXPCAPIC 00000001 BXPC 00000001)
ACPI: HPET 000000001ffffca0 00038 (v01 BOCHS  BXPCHPET 00000001 BXPC 00000001)
ACPI: MCFG 000000001ffffc60 0003C (v01 BOCHS  BXPCMCFG 00000001 BXPC 00000001)
...
ACPI: bus type pci registered
PCI: MMCONFIG for domain 0000 [bus 00-ff] at [mem 0xe0000000-0xefffffff] (base 0xe00000
PCI: MMCONFIG at [mem 0xe0000000-0xefffffff] reserved in E820
```

Hotplug slot capability

```
pciehp 0000:00:04.0:pcie04:   Physical Slot Number : 0
pciehp 0000:00:04.0:pcie04:   Attention Button     : yes
pciehp 0000:00:04.0:pcie04:   Power Controller     :  no
pciehp 0000:00:04.0:pcie04:   MRL Sensor           :  no
pciehp 0000:00:04.0:pcie04:   Attention Indicator  : yes
pciehp 0000:00:04.0:pcie04:   Power Indicator      : yes
pciehp 0000:00:04.0:pcie04:   Hot-Plug Surprise    : yes
pciehp 0000:00:04.0:pcie04:   EMI Present          : yes
pciehp 0000:00:04.0:pcie04:   Command Completed    : yes
pciehp 0000:00:04.0:pcie04: Slot Status            : 0x0000
pciehp 0000:00:04.0:pcie04: Slot Control           : 0x03c0
```

Enabled debug message via kernel command line pci_hotplug.debug=1 pci_hotplug.debug_acpi=1 pciehp.pciehp_debug=1 pci_slot.debug=1

lspci

```
# lspci -vt
-[0000:00]-+-00.0  Intel Corporation 82G33/G31/P35/P31
                   Express DRAM Controller
           +-01.0  Cirrus Logic GD 5446
...
           +-18.5-[26]--
           +-19.0-[36-bf]--+-00.0-[37-47]--+-00.0-[38]--
          ......
           |              |                +-0e.0-[46]--
           |              |                \-0f.0-[47]--
           |              +-00.1-[48-58]--+-00.0-[49]--
           |              |                +-01.0-[4a]--
           |              |                +-02.0-[4b]--
          ......
```

lspci(cont.)

```
# lspci -vvv
...
00:04.0 PCI bridge: Intel Corporation 5500 Non-Legacy I/O Hub
            PCI Express Root Port 0 (rev 02)
            (prog-if 00 [Normal decode])
        …
        Secondary status: 66MHz- FastB2B- ParErr- ...
        BridgeCtl: Parity- SERR+ NoISA- VGA- Mabort- ...
        ...
        Capabilities: [90] Express (v2) Root Port (Slot+), MSI 00
        ...
        Capabilities: [100] Advanced Error Reporting
        ...
```

Express native hot plug

- Root/downstream port is capable of hot plug.
- pci_add/pci_del, device_add/device_del
  - This is same to PCI hot plug.
  - Internally it calls back bus specific function. So it eventually pci express hotplug logic.
- pcie_apb [domain.]chassis
  - push PCI express attention buttion of a given domain and chassis number.

Hot plug

```
(qemu) pci_add 20:0 nic model=e1000
OK domain 0, bus 32, slot 0, function 0
```

```
pciehp 0000:00:04.0:pcie04: pcie_isr: intr_loc 8
pciehp 0000:00:04.0:pcie04: Presence/Notify input change
pciehp 0000:00:04.0:pcie04: Card present on Slot(0)
pciehp 0000:00:04.0:pcie04: Surprise Removal
pciehp 0000:00:04.0:pcie04: pcie_isr: intr_loc 10
pciehp 0000:00:04.0:pcie04: pciehp_green_led_blink:
    SLOTCTRL a8 write cmd 200
pciehp 0000:00:04.0:pcie04: pciehp_check_link_status:
    lnk_status = 11
...
e1000 0000:20:00.0: enabling device (0000 -> 0003)
...
```

Push attention button

```
 (qemu) pcie_abp 0
 OK chassis 0, slot 0

pciehp 0000:00:04.0:pcie04: pcie_isr: intr_loc 1
pciehp 0000:00:04.0:pcie04: Attention button interrupt received
pciehp 0000:00:04.0:pcie04: Button pressed on Slot(0)
...
pciehp 0000:00:04.0:pcie04: PCI slot #0 -
    powering off due to button press.
...
pciehp 0000:00:04.0:pcie04: pciehp_unconfigure_device:
    domain:bus:dev = 0000:20:00
e1000: eth1: e1000_reset: Hardware Error
e1000 0000:20:00.0: PCI INT A disabled
```

Hot unplug

```
(qemu) pci_del 20:0

pciehp 0000:00:04.0:pcie04: pcie_isr: intr_loc 8
pciehp 0000:00:04.0:pcie04: Presence/Notify input change
pciehp 0000:00:04.0:pcie04: Card not present on Slot(0)
pciehp 0000:00:04.0:pcie04: Surprise Removal
pciehp 0000:00:04.0:pcie04:
    Disabling domain:bus:device=0000:20:00
pciehp 0000:00:04.0:pcie04:
    pciehp_unconfigure_device: domain:bus:dev = 0000:20:00
pciehp 0000:00:04.0:pcie04: pcie_isr: intr_loc 10
pciehp 0000:00:04.0:pcie04:
    pciehp_green_led_off: SLOTCTRL a8 write cmd 300
```

AER: Error injection

- pcie_error_inject [[domain:]bus:]dev.fn is_correctable error_status number number number number [number [number [number [number]]]]
  - is_correctable: bool
    - Correctable or uncorrectable
  - error_status: uint32_t
    - Spcify error type
  - number: uint32_t*4: TLP header
  - number: uint32_t*{0-4}: TLP header prefix

```
    (qemu)    pcie_aer_inject_error 0:0:4.0 0x10
    msi_notify:295 pcie-root-port:20 notify vector 0x0
       address: 0xfee0100c data: 0x4129
    OK domain: 0, bus: 0 devfn: 4.0

pcieport 0000:00:04.0: AER: Uncorrected (Non-Fatal)
    error received: id=0020
pcieport 0000:00:04.0: PCIe Bus Error: severity=Uncorrected
    (Non-Fatal), type=Transaction Layer, id=0020(Receiver ID)
pcieport 0000:00:04.0:   device [8086:3420] error
    status/mask=00001000/00000000
pcieport 0000:00:04.0:    [12] Poisoned TLP
pcieport 0000:00:04.0:   TLP Header: 00000000 00000000 00000000
     00000000
pcieport 0000:00:04.0: broadcast error_detected message
pcieport 0000:00:04.0: broadcast mmio_enabled message
pcieport 0000:00:04.0: broadcast resume message
pcieport 0000:00:04.0: AER driver successfully recovered
```

Future work

Future Work

- Upstream merge
- PCI express native device assignment
  - PCI express specific configuration registers should be virtualized
    - Device serial number cap, VSEC...
  - AER(Advanced Error Report)
    - Catch the error in host.
      - Currently Linux AER port driver does only printk().
      - Poll errors from targeted devices.
    - inject errors from host to guest OS for RAS.
  - Native Power management
  - VC(Virtual channel)
  - Assigning bus hierarchy tree
- Multifunction hot plug

Future work: device assignment support

- Support direct device assignment in qemu witout kvm, xen?
- Hopefully consolidate kvm and xen passthrough code into qemu.
- By consolidating the passthrough code into qemu, the code base would get more tests and become more stable.

Future work: IOMMU

- Real Q35 has VT-d.
- Qemu iommu(Intel VT-d, AMD IOMMU) emulation is coming.
  - So device assignment version would be wanted.
- Shadowing IOMMU page tables for guest OS
  - For nested virtualization

Summary

- PCI Express is useful even in virtualized environment
- Q35 new chipset patch enables QEmu to support PCI Express
- It benefits all qemu derivatives, KVM and Xen.

Thank you

Questions?